Historical document image analysis : a structural approach based on texture

Maroua Mehri

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE S2IM

THÈSE présentée par :

Maroua MEHRI

préparée aux : **Laboratoire Informatique, Image et Interaction (L3i)**
&
Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS)

soutenue le : **28 mai 2015**
pour obtenir le grade de : **Docteur de l'Université de La Rochelle**

Discipline : **Informatique et applications**

Analyse d'images de documents patrimoniaux : une approche structurelle à base de texture

---

JURY :

Najoua ESSOUKRI BEN AMARA	Professeur, Université de Sousse (Tunisie), Examinateur, Président du jury
Rolf INGOLD	Professeur, Université de Fribourg (Suisse), Rapporteur
Josep LLADÒS	Professeur associé, Université Autonome de Barcelone (Espagne), Rapporteur
Véronique EGLIN	Maître de conférences, HDR, INSA de Lyon (France), Examinateur
Jean-Philippe MOREUX	Chef de projet OCR, Bibliothèque Nationale de France (France), Invité
Rémy MULLOT	Professeur, Université de La Rochelle (France), Directeur de thèse
Pierre HÉROUX	Maître de conférences, Université de Rouen (France), Encadrant de thèse
Petra GOMEZ-KRÄMER	Maître de conférences, Université de La Rochelle (France), Encadrant de thèse

UNIVERSITY OF LA ROCHELLE

S2IM DOCTORAL SCHOOL

THESIS by:

Maroua MEHRI

carried out at: **Laboratoire Informatique, Image et Interaction (L3i)**
&
Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS)

defended on: **28 May 2015**
for the award of the degree of: **Doctor of Philosophy of University of La Rochelle**
Discipline: **Computer science and applications**

Historical document image analysis: a structural approach based on texture

---

JURY:

Najoua ESSOUKRI BEN AMARA	Professor, University of Sousse (Tunisia), Examiner, Committee chair
Rolf INGOLD	Professor, University of Fribourg (Switzerland), Reviewer
Josep LLADÒS	Associate professor, Universitat Autònoma de Barcelona (Spain), Reviewer
Véronique EGLIN	Associate professor, INSA Lyon (France), Examiner
Jean-Philippe MOREUX	OCR project leader, French National Library (France), Invited expert
Rémy MULLOT	Professor, University of La Rochelle (France), Thesis director
Pierre HÉROUX	Associate professor, University of Rouen (France), Thesis supervisor
Petra GOMEZ-KRÄMER	Associate professor, University of La Rochelle (France), Thesis supervisor

This document was typeset by the author using LaTeX.

The jointly supervised research described in this book was carried out at the "laboratoire informatique, image et interaction" (L3i) from the University of La Rochelle, and at the "laboratoire d'informatique, du traitement de l'information et des systèmes" (LITIS) from the University of Rouen.

Printed by L3i-University of La Rochelle.

Dedicated to my parents...

Acknowledgements

First and foremost I would like to express my deepest gratitude to who have helped and supported me.

Sincere thanks to my supervisors Pr. Dr. Rémy Mullot, Dr. Pierre Héroux and Dr. Petra Gomez-Krämer for their valuable support, encouragement, review of this thesis and joint supervision of this work between the "*laboratoire informatique, image et interaction*" (L3i) from the University of La Rochelle and the "*laboratoire d'informatique, du traitement de l'information et des systèmes*" (LITIS) from the University of Rouen. My discussions with them have been interesting and inspiring.

I also want to thank my committee members. I am grateful to Pr. Dr. Rolf Ingold and Pr. Dr. Josep Lladòs to have accepted to review this thesis. I am also grateful to Dr. Véronique Eglin and Eng. Jean-Philippe Moreux for accepting to be part of the jury as examiners and Pr. Dr. Najoua Essoukri Ben Amara to have presided the jury.

I would also thank Pr. Dr. Jean-Marc Ogier and Pr. Dr. Thierry Paquet for accepting me in their laboratories and both institutions for giving me the opportunity to work in a collaborative working environment, a great rewarding working atmosphere and cheerfulness. I would like to thank the DIGIDOC project founds for the financial support of this work. Thank you to the DIGIDOC project's team for all the fruitful meetings we had together. My work was supported by the French national research agency (ANR), under Grant ANR-10-CORD-0020, which is gratefully acknowledged. I would like also to thank Geneviève Cron and Christos Papadopoulos for providing access to the Gallica digital library and IMPACT dataset, respectively.

I want also to thank Kathy Theuil, Geneviève Chiali, Fabienne Bocquet, Chantal Le Maistre, Dominique Joffroy, Marie-Chrystel Gobin, Erlandri Chaigneaud, Sarah Ehlinger, Caroline Bourmaud, Hervé Tichané, Jennifer De La Corte Gomez, Isabelle Hirsch and Francoise Meric de Bellefon for their help with the administrative stuff. Grateful thanks to Dominique Limousin, Mikael Guichard, Romain Vandebogarde and Fabrice Hertel for supplying me all necessary equipment to carry out my work. Thank you to Stéphane Djerdjar, samuel Laborde and Annick Mercier for their good humor and kindness.

I would also like to express special thanks to my thesis co-adviser Dr. Pierre Héroux for having welcomed me on several occasions in LITIS lab and taking the time to answer all my questions and for being helpful. Kind thanks to Dr. Nibal Nayef, Dr. Sophea Prum, Pr. Dr. Sébastien Adam, Julien Lerouge, Vincent Rabeux, Elodie Carel, Romain Bertrand, Gaël Le Baccon, Antoine Mercier, Dr. Véronique Eglin, Dr. Muriel Visani, Dr. Nicholas Journet, Dr. Mickaël Coustaty, Dr. Thomas Martin and Dr. Cyril Faucher for sharing useful feedback and interesting comments and discussions. Thank you to Dr. Nicholas Journet, Dr. Muriel Visani, Pr. Dr. Jean-Philippe Domenger, Dr. Van Cuong Kieu, Pr. Dr. Najoua Essoukri Ben Amara and Dr. Mohamed Ali Mahjoub for giving me the great pleasure of collaborating with the two laboratories, "*laboratoire Bordelais de recherche en informatique*" (LaBRI) and "*systèmes avancés en génie électrique*" (SAGE). Thanks to the trainees, Mohamed Mhiri and Nabil Sliti, that contributed to this work.

I would like to thank Dr. Laurent Albera, Dr. Pierre Jannin, Dr. Florent Lalys and Dr. Mohamed Lassaad Ammari that gave me the taste of academic research of signal and image processing during the research master final year project at the University of Rennes 1 and the University of Sousse. Thank you to the people who have provided me the opportunity to teach during the last three years in the computer science department of the institute of technology (IUT) at the University of La Rochelle (Philippe Coulaud, Philippe Crottereau, Dr. Ronan Champagnat, Dr. Farid Ammar-Boudjelal, Laurent Berndt, Dr. Petra Gomez-Krämer, Pr. Dr. Mohamed Yacine Ghamri-Doudane and Dr. Jamal Malki).

Thank you to all the Ph.D. candidates, doctors, engineers and trainees affiliated to the two laboratories, L3i and LITIS, with whom I spend most of the time. Special thanks to the 123 open space team (Dr. Sophea Prum, Imen Bizid, Marcela Rojas Castro, Wong Poh Lee, Hind Idrissi, Dr. Cyril Faucher, Dr. Thomas Martin, Gaël Le Baccon) and Sovann En for the good moments spent together in both the L3i and LITIS offices. Grateful thanks to my friends in Sousse, Rennes and La Rochelle for their great encouragement. And finally, I would like to infinitely all my family and particularly my father for his tremendous help and support.

Abstract

Over the last few years, there has been tremendous growth in digitizing collections of cultural heritage documents. Thus, many challenges and open issues have been raised, such as information retrieval in digital libraries or analyzing page content of historical books. Recently, an important need has emerged which consists in designing a computer-aided characterization and categorization tool, able to index or group historical digitized book pages according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the historical document image content.

Current systems for categorizing historical digitized book pages are based on several criteria, such as the textual content. However, these systems for performing the historical document image analysis tasks have poor performance due to many particularities of historical document images (e.g. large variability of page layout, noise and degradation, page skew, complicated layout, random alignment, specific fonts, presence of embellishments, variations in spacing between the characters, words, lines, paragraphs and margins, overlapping object boundaries, superimposition of information layers). Moreover, these systems are hindered by many issues related to the performance of the optical character recognition and retrospective conversion tools. In addition, they require burdensome and complex processing due to the mentioned particularities of historical document images.

Thus, the work conducted in this thesis presents an automatic approach for characterization and categorization of historical book pages. The proposed approach is applicable to a large variety of ancient books. In addition, it does not assume *a priori* knowledge regarding document image layout and content. It is based on the use of texture and graph algorithms to provide a rich and holistic description of the layout and content of the analyzed book pages to characterize and categorize historical book pages. The categorization is based on the characterization of the digitized page content by texture, shape, geometric and topological descriptors. This characterization is represented by a structural signature. More precisely, the categorization consists of two main stages. The first stage is extracting homogeneous regions. Then, the second one is proposing a graph-based page signature which is based on the extracted homogeneous regions, reflecting its layout and content.

First, a bottom-up segmentation approach based on analyzing texture features which have been extracted using a multi-scale analysis technique, has been performed for identifying homogeneous regions. Given that there are significant degradations and no hypothesis concerning the layout, the graphical properties or typographical parameters of historical document images, the use of a texture-based approach has become an appropriate choice. Indeed, the proposed texture-based approach addresses the needs for segmenting a page *(i)* under significant degradations and different noise levels and types, *(ii)* without *a priori* knowledge regarding page layout and content.

Once, the homogeneous regions have been extracted, the second stage of the proposed approach consists in constructing a structural representation (*i.e.* a graph-based signature). The graph vertices correspond to the extracted homogeneous regions. Each vertex is described by texture, shape, geometric and topological descriptors, characterizing the region. On the other hand, a set of edges is built based on topological relationships connecting the different extracted homogeneous regions.

Afterwards, by comparing the different obtained graph-based signatures using a graph-matching paradigm, the similarities of digitized historical book page layout and/or content can be deduced. Subsequently, book pages with similar layout and/or content can be categorized and grouped, and a table of contents/summary of the analyzed digitized historical book can be provided automatically.

This structural signature combining the layout and content description, ensures the characterization of historical document image book. Thus, it provides several possible operational and interesting options of categorization, indexing and retrieval of digitized resources. In addition, it offers a structured multi-criteria access to large sets of cultural heritage documents, without using the optical character recognition and retrospective conversion tools and with as little *a priori* knowledge as possible. Indeed, numerous signature-based applications (e.g. information retrieval in digital libraries according to several criteria, page categorization) can be implemented for managing effectively a corpus or collections of books.

In this dissertation, we have investigated how this structural signature ensures the design of a computer-aided characterization and categorization approach, able to compare or group digitized historical book pages according to several criteria, mainly the layout structure, graphical characteristics or typographic properties of the historical document image content. To illustrate the effectiveness of the proposed page signature, a detailed experimental evaluation has been conducted in this work for assessing two possible categorization applications, unsupervised page classification and page stream segmentation. In addition, the different steps of the proposed approach have been evaluated on a large variety of historical document images.

Résumé

Les récents progrès dans la numérisation des collections de documents patrimoniaux ont ravivé de nouveaux défis afin de garantir une conservation durable et de fournir un accès plus large aux documents anciens. En parallèle de la recherche d'information dans les bibliothèques numériques ou l'analyse du contenu des pages numérisées dans les ouvrages anciens, la caractérisation et la catégorisation des pages d'ouvrages anciens a connu récemment un regain d'intérêt. Les efforts se concentrent autant sur le développement d'outils rapides et automatiques de caractérisation et catégorisation des pages d'ouvrages anciens, capables de classer les pages d'un ouvrage numérisé en fonction de plusieurs critères, notamment la structure des mises en page et/ou les caractéristiques typographiques/graphiques du contenu de ces pages.

Les systèmes actuels de caractérisation et catégorisation des pages d'ouvrages numérisés s'appuient sur plusieurs critères relatifs au contenu textuel. Cependant, des performances insatisfaisantes ont été relevées en raison de divers problèmes, et qui sont liés aux particularités des documents anciens (e.g. une grande variabilité de la mise en page, des niveaux différents de dégradation et bruit, le défaut d'orientation, la complexité de la mise en page, des alignements non-conventionnels, les polices de caractères spécifiques, la présence d'ornements, les variations de l'espacement entre les caractères, mots, lignes, paragraphes et marges, la superposition de plusieurs couches d'information). En effet, leurs performances sont étroitement liées à celles des outils de reconnaissance optique de caractères et rétro-conversion. En outre, le traitement de ce type de documents peut s'avérer complexe et pénible en raison des particularités des documents anciens mentionnées ci-dessus, et ce, sans connaissances *a priori* sur la structure des mises en page ou les caractéristiques typographiques/graphiques du contenu de ces pages.

Ainsi, dans le cadre de cette thèse, nous proposons une approche permettant la caractérisation et la catégorisation automatiques des pages d'un ouvrage ancien. L'approche proposée se veut indépendante de la structure et du contenu de l'ouvrage analysé. Le principal avantage de ce travail réside dans le fait que l'approche s'affranchit des connaissances préalables, que ce soit concernant le contenu du document ou sa structure. Elle est basée sur une analyse des descripteurs de texture et une représentation structurelle en graphe afin de fournir une description riche permettant une catégorisation à partir du contenu graphique (capturé par la texture) et des mises en page (représentées par des graphes). En effet, cette catégorisation s'appuie sur la caractérisation du contenu de la page numérisée à l'aide d'une analyse des descripteurs de texture, de forme, géométriques et topologiques. Cette caractérisation est définie à l'aide d'une représentation structurelle. Dans le détail, l'approche de catégorisation se décompose en deux étapes principales successives. La première consiste à extraire des régions homogènes. La seconde vise à proposer une signature structurelle à base de texture, sous la forme d'un graphe, construite à partir des régions homogènes extraites et reflétant la structure de la page analysée. Cette signature assure la mise en œuvre de nombreuses applications pour gérer efficacement un corpus ou des collections de livres patrimoniaux (e.g. la recherche d'information dans les bibliothèques numériques en fonction de plusieurs critères, la catégorisation des pages d'un même ouvrage). En comparant les différentes signatures structurelles par le biais de la distance d'édition entre graphes, les similitudes entre les pages d'un même ouvrage en termes de leurs mises en page et/ou contenus peuvent être déduites. Ainsi de suite, les pages ayant des mises en page et/ou contenus similaires peuvent être catégorisées, et un résumé/une table des matières de l'ouvrage analysé peut être alors généré automatiquement.

En effet, une approche ascendante de segmentation exploitant des descripteurs de texture mesurés à différentes échelles est tout d'abord proposée pour l'extraction des régions homogènes. Cette approche est notamment guidée par *(i)* la nécessité de robustesse au bruit fréquemment présent sur les images de documents anciens, *(ii)* le fait de pouvoir traiter des documents dont les mises en page et caractéristiques typographiques sont variées et, *a priori*, inconnues.

Dès lors que les zones homogènes ont été extraites, la seconde étape de l'approche construit une signature structurelle de la page (*i.e.* graphe). Les nœuds du graphe ainsi produits sont associés aux zones homogènes et sont étiquetés par les attributs caractérisants les régions. Les arcs, quant à eux, caractérisent les liens topologiques entre les différentes régions.

Cette signature structurelle associant représentation des éléments de contenu et description de la mise en page, caractérise les pages de documents anciens numérisés à différents niveaux. Elle offre ainsi plusieurs modalités de catégorisation et d'indexation permettant une navigation multi-critère dans les corpus, et ce, sans reconnaissance et en ayant introduit aussi peu de connaissances *a priori* que possible. Dans le cadre de cette thèse, nous avons notamment étudié comment la signature produite par l'approche proposée pouvait être exploitée afin de comparer et catégoriser les pages d'un même ouvrage. Pour illustrer l'efficacité de la signature proposée, une étude expérimentale détaillée a été menée dans ce travail pour évaluer deux applications possibles de catégorisation de pages d'un même ouvrage, la classification non supervisée de pages et la segmentation de flux de pages d'un même ouvrage. En outre, les différentes étapes de l'approche proposée ont donné lieu à des évaluations par le biais d'expérimentations menées sur un large corpus de documents patrimoniaux.

Table of contents

List of Figures

List of Tables

List of Algorithms

Notations

- $\boldsymbol{I}$: input image
- $\boldsymbol{W}$: width of an image
- $\boldsymbol{H}$: height of an image
- $\boldsymbol{S}$: size of an image
- $\boldsymbol{E}$: structuring element of a morphology-based method
- $\oplus$: dilation operator
- $\ominus$: erosion operator
- $\circ$: opening operator
- $\bullet$: closing operator
- $\boldsymbol{d_e}$: distance between each k-NN pair of the extracted CCs in the docstrum algorithm
- $\boldsymbol{\Phi_e}$: angle of each edge in the docstrum algorithm
- $\boldsymbol{P_V = \{p_1, \ldots, p_n\}}$: point set or generators of the Voronoi diagram
- $\boldsymbol{d(p,q)}$: distance between points p and q
- $\boldsymbol{\bar{V}(p_i)}$: Voronoi region
- $\boldsymbol{V(P_V)}$: Voronoi diagram
- $\boldsymbol{F^i}$: texture feature
- $\boldsymbol{\bar{I}(x,y)}$: image pixel
- $\boldsymbol{f(x,y)}$: gray-level of image pixel
- $\boldsymbol{\mu}$: mean value
- $\boldsymbol{\mu_4}$: fourth moment
- $\boldsymbol{\sigma}$: standard deviation estimator
- $\boldsymbol{k_t}$: neighborhood size at image pixel $I(x, y)$ such as the size of the analysis image is equal to $2^{k_t} \times 2^{k_t}$
- $\boldsymbol{A_{k_t}(x,y)}$: computed average for the windows of size $2^{k_t} \times 2^{k_t}$ to estimate the coarseness feature
- $\boldsymbol{E_{k_t,h}(x,y)}$: difference between the average of pairs corresponding to pairs of non-overlapping neighborhoods on opposite sides of the analyzed pixel in both the horizontal orientation
- $\boldsymbol{E_{k_t,v}(x,y)}$: difference between the average of pairs corresponding to pairs of non-overlapping neighborhoods on opposite sides of the analyzed pixel in both the vertical orientation
- $\boldsymbol{S_{best}}$: sequence for the estimation of the coarseness feature

- $\boldsymbol{Hist_D}$: histogram of local edge probabilities
- $\boldsymbol{\nabla_H}$: horizontal mask for the estimation of the number of orientations
- $\boldsymbol{\nabla_V}$: vertical mask for the estimation of the number of orientations
- $\boldsymbol{|\Delta G|}$: magnitude for the edge detection
- $\boldsymbol{\theta_t}$: direction for the edge detection
- $\boldsymbol{t_{Hist}}$: specified $Hist_D$ threshold
- $\boldsymbol{n_b}$: number of the $Hist_D$ bins
- $\boldsymbol{N_{\theta_t}(k)}$: number of pixels for the estimation of the number of orientations
- $\boldsymbol{n_p}$: number of histogram peaks
- $\boldsymbol{\Phi_p}$: p^{th} peak position of $Hist_D$
- $\boldsymbol{w_p}$: range of p^{th} peak between valleys
- $\boldsymbol{r}$: normalizing factor related to the quantized levels of Φ_h
- $\boldsymbol{\Phi_h}$: quantized direction code (cyclically in modulo 180°)
- $\boldsymbol{I_c(x,y)}$: analyzed image pixel
- $\boldsymbol{I_p(x,y)}$: image pixels defined in the P circularly symmetric neighbors
- $\boldsymbol{f_c(x,y)}$: gray-level of the analyzed image pixel $I_c(x,y)$
- $\boldsymbol{f_p(x,y)}$: gray-level of the image pixel $I_p(x,y)$
- $\boldsymbol{P_l}$: number of neighboring pixels in a circular set
- $\boldsymbol{R_l}$: radius of a circular set
- $\boldsymbol{n_l}$: number of the unique rotation invariant local binary patterns
- $\boldsymbol{Hist_{P_l,R_l}}$: histogram of binary patterns
- $\boldsymbol{LBP_{P_l,R_l}}$: LBP operator
- $\boldsymbol{LBP^{ri}_{P_l,R_l}}$: rotation invariant LBP operator
- $\boldsymbol{LBP^{u2}_{P_l,R_l}}$: uniform 2 LBP operator
- $\boldsymbol{LBP^{riu2}_{P_l,R_l}}$: rotation invariant uniform 2 LBP operator
- $\boldsymbol{Hist_{g,l}}$: histogram of run-lengths
- $\boldsymbol{g}$: gray-level value bin
- $\boldsymbol{G^l}$: number of gray-level bins
- $\boldsymbol{l}$: run-length
- $\boldsymbol{L}$: maximum run-length
- $\boldsymbol{\theta_r}$: scan direction of a GLRLM gray-level run

- $\boldsymbol{p(g,l)}$**:** element of the GLRLM
- $\boldsymbol{\bar{P}(g,l)}$**:** probability of a specific run-length
- $\boldsymbol{I(x+\alpha, y+\beta)}$**:** translation of the analysis window of an image $I(x,y)$ by α and β pixels along the horizontal and vertical axes, respectively, defined on the plane Ω
- $\boldsymbol{R^{I(\alpha,\beta)}_{(x,y)}}$**:** auto-correlation function computed along the horizontal and vertical axes of the analysis window of an image I
- $\boldsymbol{FFT}$**:** fast Fourier transform
- $\boldsymbol{(.)^*}$**:** complex conjugate
- $\boldsymbol{(.)^{-1}}$**:** inverse transform
- $\boldsymbol{\Theta_i}$**:** selected orientation
- $\boldsymbol{D_i}$**:** set of possible orientations
- $\boldsymbol{R^I_{min}}$**:** minimum value of $R^I_{(x,y)}(\Theta_i)$
- $\boldsymbol{\bar{R}^I_{max}}$**:** maximum values of $R^I_{(x,y)}(\Theta_i)$
- $\boldsymbol{R'^I_{(x,y)}(\Theta_i)}$**:** relative sum of the different values of the auto-correlation function
- $\boldsymbol{\theta_a}$**:** number of orientation values of the rose of directions
- $\boldsymbol{S^{width}}$**:** sequence for the estimation the mean stroke width
- $\boldsymbol{T^{\Theta}_{(\alpha,0)}(I(.,.))}$**:** translation of the analysis window of an image I by α pixels along the axis of the main angle of the rose of directions
- $\boldsymbol{S^{height}}$**:** sequence for the estimation the mean stroke height
- $\boldsymbol{T^{\Theta}_{(0,\beta)}(I(.,.))}$**:** translation of the analysis window of an image I by β pixels along the axis of the main angle of the rose of directions
- $\boldsymbol{\theta_c}$**:** specified direction of the GLCM calculation
- $\boldsymbol{d_c}$**:** specified distance of the GLCM calculation
- $\boldsymbol{p_{d_c,\theta_c}(i,j)}$**:** probability of the gray-level pair i and j defined in a specified direction θ_c and separated by a particular distance of d_c units
- $\boldsymbol{f_g}$**:** spatial frequency of the Gabor filter envelope
- $\boldsymbol{\theta_g}$**:** orientation of the Gabor filter envelope
- $\boldsymbol{\sigma_g}$**:** space constant of the Gabor filter envelope
- $\boldsymbol{I_{G_{(f_g,\theta_g)}}(x,y)}$**:** Gabor filtered image of an image $I(x,y)$
- $\boldsymbol{G_{(f_g,\theta_g)}(\alpha,\beta)}$**:** spatial frequency response of Gabor filter
- $\boldsymbol{G_{e(f_g,\theta_g)}}$**:** spatial frequency response of the even-symmetric Gabor filter
- $\boldsymbol{G_{o(f_g,\theta_g)}}$**:** spatial frequency response of the odd-symmetric Gabor filter
- $\boldsymbol{M_g}$**:** width of the Gabor filtered magnitude response

- N_g: height of the Gabor filtered magnitude response
- g^f: high-pass filter for 2D wavelet decomposition
- h^f: low-pass filter for 2D wavelet decomposition
- $Haar$: 3-level Haar wavelet transform
- $Db3$: 3-level wavelet transform using 3-tap Daubechies filter
- $Db4$: 3-level wavelet transform using 4-tap Daubechies filter
- ϕ: 2D scaling function
- ψ: wavelet function
- J: scale of the discrete wavelet transform
- j: decomposition level of the discrete wavelet transform
- $A_{2^{-J}}$: approximation of the input image at 2^{-J} resolution
- $D^{(v)}_{2^{-j}}$: vertical detail components of the input image at 2^{-j} resolution
- $D^{(h)}_{2^{-j}}$: horizontal detail components of the input image at 2^{-j} resolution
- $D^{(d)}_{2^{-j}}$: diagonal detail components of the input image at 2^{-j} resolution
- $C^{Aj}_{k,l}$: approximation coefficients at 2^{-j} resolution
- $C^{D(s)j}_{k,l}$: detail coefficients at 2^{-j} resolution
- $(s)j$: vertical, horizontal or diagonal detail components of the input image at 2^{-j} resolution
- $f_s(x,y)$: pixel gray-level of a sub-band or sub-image from the 2D wavelet decomposition
- $C(i,j)$: transform wavelet coefficient
- S_w: width of a sub-band in the wavelet domain
- S_h: height of a sub-band in the wavelet domain
- g^f_{Haar}: high-pass filter of the Haar wavelet transform
- h^f_{Haar}: low-pass filter of the Haar wavelet transform
- g^f_{Db3}: high-pass filter of the Db3 wavelet transform
- h^f_{Db3}: low-pass filter of the Db3 wavelet transform
- g^f_{Db4}: high-pass filter of the Db4 wavelet transform
- h^f_{Db4}: low-pass filter of the Db4 wavelet transform
- N^f: number of extracted textural indices by applying multi-scale analysis
- V^f: feature vector
- x_i: element of the feature vector V^f

- $\Re$: real
- $\overline{x_{ak}}$: centroid of cluster a
- $\overline{x_{bk}}$: centroid of cluster b
- n_a: number of elements in cluster a
- n_b: number of elements in cluster b
- k: number of clusters
- I_t: number of extracted Tamura indices
- I_l: number of extracted LBP indices
- I_r: number of extracted GLRLM indices
- I_a: number of extracted auto-correlation indices
- $\bar{I}_c$: number of extracted GLCM indices
- $\bar{I}_g$: number of extracted Gabor indices
- $\bar{I}_h$: number of extracted Haar indices
- $\bar{I}_{db3}$: number of extracted Db3 indices
- $\bar{I}_{db4}$: number of extracted Db4 indices
- $I_{A_{2^{-J}}}$: number of extracted approximation sub-image indices
- $I_{D^{(v)}_{2^{-j}}}$: number of extracted vertical detail sub-image indices
- $I_{D^{(h)}_{2^{-j}}}$: number of extracted horizontal detail sub-image indices
- $I_{D^{(d)}_{2^{-j}}}$: number of extracted diagonal detail sub-image indices
- N_w: number of sliding windows
- n_r: number of pixels of the sliding window
- M: number of foreground pixels
- n_g: number of gray-levels
- n_t: number of averages $A_{k_t}(x, y)$ for the windows of size $2^{k_t} \times 2^{k_t}$
- $.'$: . minutes
- $.''$: . seconds
- SED: squared Euclidean distance
- WED: weighted Euclidean distance
- SW: average silhouette width
- x_i: cluster point
- $SW(x_i)$: silhouette width for each point x_i

- $\boldsymbol{a(x_i)}$**:** compactness between x_i and the other points in the same cluster
- $\boldsymbol{b(x_i)}$**:** separation between x_i and the closest cluster
- $\boldsymbol{K(x_i)}$**:** cluster containing the point x_i
- $\boldsymbol{D(x_i, x_j)}$**:** distance between two points x_i and x_j
- $\boldsymbol{K_l}$**:** cluster that does not contain the point x_i
- $\boldsymbol{N_l}$**:** number of points in the cluster K_l
- $\boldsymbol{N}$**:** number of points in the dataset
- $\boldsymbol{J}$**:** Jaccard coefficient
- $\boldsymbol{N_{11}}$**:** number of pairs of data points which are clustered together in the clustering result and ground-truth
- $\boldsymbol{N_{10}}$**:** number of pairs of data points which are clustered together in the clustering result but not in the ground-truth
- $\boldsymbol{N_{01}}$**:** number of pairs of data points which are clustered together in the ground-truth but not in the clustering result
- $\boldsymbol{PPB}$**:** purity per block metric
- $\boldsymbol{|.|}$**:** number of pixels in a given block
- $\boldsymbol{B}$**:** set of result blocks
- $\boldsymbol{b_i}$**:** result block
- $\boldsymbol{G^t}$**:** set of rectangular regions of the ground-truth
- $\boldsymbol{g_j^t}$**:** pre-defined rectangular region of the ground-truth
- $\boldsymbol{L_B}$**:** set of labels obtained with the used pixel clustering technique
- $\boldsymbol{l_{B_i}}$**:** label corresponding to the result block obtained with the used pixel clustering technique
- $\boldsymbol{M_c}$**:** confusion matrix, error matrix or contingency table
- $\boldsymbol{E}$**:** entropy
- $\boldsymbol{PT}$**:** purity
- $\boldsymbol{P}$**:** precision
- $\boldsymbol{R}$**:** recall
- $\boldsymbol{CA}$**:** classification accuracy rate
- $\boldsymbol{F}$**:** F-score or F-measure
- $\boldsymbol{c}$**:** set of classes in the dataset
- $\boldsymbol{c_i}$**:** dataset class
- $\boldsymbol{Pr_i(c_j)}$**:** proportion of the data point class c_j in the cluster i

- $\boldsymbol{N_d}$**:** number of the M_c diagonal elements which represent the all correctly assigned samples to theirs classes
- $\boldsymbol{N_o}$**:** number of the M_c elements, excluding those of its diagonal, along a column (clustering outcomes) correspond to omission samples
- $\boldsymbol{N_c}$**:** number of the M_c elements, excluding those of its diagonal, along a row (ground-truth classes) correspond to commission samples
- $\boldsymbol{n}$**:** order of the square confusion matrix M_c
- $\boldsymbol{m_{pq}}$**:** number of elements of class q assigned to cluster p
- $\boldsymbol{P_i}$**:** precision of the cluster i
- $\boldsymbol{R_j}$**:** recall of the class j
- $\boldsymbol{M_{mc}}$**:** merge consensus matrix
- $\boldsymbol{k_{opt}}$**:** optimal number of clusters
- $\boldsymbol{CDF(c)}$**:** cumulative density function
- $\boldsymbol{N_s}$**:** number of selected observations or samples
- $\mathbb{1}$**:** indicator or a characteristic function
- $\boldsymbol{AUC}$**:** area under the cumulative density curve
- $\boldsymbol{y_i}$**:** current element of the CDF
- $\boldsymbol{m}$**:** number of elements of the CDF
- $\boldsymbol{\Delta k}$**:** difference change between two consecutive elements k in the AUC
- $\boldsymbol{MD}$**:** Mahalanobis distance
- $\boldsymbol{S}$**:** covariance matrix
- $\boldsymbol{.\text{-}D}$**:** .-dimensional
- $\boldsymbol{N^f\text{-}D}$**:** N^f-dimensional
- $\boldsymbol{GB}$**:** gigabytes
- $\boldsymbol{MB}$**:** megabytes
- $\boldsymbol{k_{est}}$**:** estimated number of clusters
- $\boldsymbol{k_{gt}}$**:** number of clusters defined in the ground-truth
- $\boldsymbol{D_k(k_{est}, k_{gt})}$**:** difference between the number of clusters *vs.* classes
- $\boldsymbol{Image_b}$**:** binarized document image
- $\boldsymbol{Image_{ref}}$**:** refined pixel-labeled document image
- $\boldsymbol{Image_{mv}}$**:** resulting document image derived from the application of the majority voting technique
- $\boldsymbol{Image^l_{mv}}$**:** resulting document image derived from the application of the color layer separation task

- $\boldsymbol{Image_{b^l_{mv}}}$**:** binarized document image derived from the binarization of the resulting document image of the application of the color layer separation task
- $\boldsymbol{Image^l}$**:** binarized document image derived from the application of the logical NOT on the $Image_{b^l_{mv}}$
- $\boldsymbol{Image_{b,post}}$**:** binarized post-processed document image
- $\boldsymbol{Image_{post}}$**:** post-processed pixel-labeled document image
- $\boldsymbol{CC_b}$**:** extracted CCs from the $Image_b$
- $\boldsymbol{CC_{ref}}$**:** extracted CCs from the $Image_{ref}$
- $\boldsymbol{CC_{mv}}$**:** extracted CCs from the $Image_{mv}$
- $\boldsymbol{CC_{b^l_{mv}}}$**:** extracted CCs from the $Image_{b^l_{mv}}$
- $\boldsymbol{CC_{post}}$**:** extracted CCs from the $Image_{post}$
- $\boldsymbol{CC^{rep}_{post}}$**:** selected representative homogeneous regions from CC_{post}
- $\boldsymbol{Image^h_{RLSA}}$**:** resulting document image derived from the application of the RLSA algorithm on the $Image_{b^l_{mv}}$ in the horizontal direction with the estimated horizontal run-length smoothing value (T_h)
- $\boldsymbol{Image^v_{RLSA}}$**:** resulting document image derived from the application of the RLSA algorithm on the $Image_{b^l_{mv}}$ in the vertical direction with the estimated vertical run-length smoothing value (T_v)
- $\boldsymbol{CC^i}$**:** number of the extracted CC_{post}
- $\boldsymbol{S_{CC^i}}$**:** number of pixels belonging to the CC^i
- $\boldsymbol{S_{CC_{post}}}$**:** total number of pixels of all extracted CC_{post}
- $\boldsymbol{ED}$**:** Euclidean distance
- $\boldsymbol{V_{p_f}}$**:** Gabor feature vector of the selected foreground pixel
- $\boldsymbol{V^c_{p_f}}$**:** Gabor feature vector of the centroid of cluster belonging to the selected foreground pixel
- $\boldsymbol{T_h}$**:** estimated horizontal threshold ARLSA
- $\boldsymbol{T_v}$**:** estimated vertical threshold ARLSA
- $\boldsymbol{GMH_w}$**:** global maximum of the histogram of the widths of the extracted CCs
- $\boldsymbol{GMH_h}$**:** global maximum of the histogram of the heights of the extracted CCs
- $\boldsymbol{T_c}$**:** pre-defined threshold used to exclude the CCs corresponding to noise
- $\boldsymbol{c_h}$**:** pre-defined weight used for computing the horizontal threshold T_h
- $\boldsymbol{c_v}$**:** pre-defined weight used for computing the vertical threshold T_v
- $\boldsymbol{\mathscr{G}}$: set of graphs
- $\boldsymbol{G}$**:** graph
- $\boldsymbol{G_v}$**:** graph vertices

- $\boldsymbol{G_e}$**:** graph edges
- $\boldsymbol{v^i}$**:** graph vertex
- $\boldsymbol{e^i}$**:** graph edge
- $\boldsymbol{o_i}$**:** elementary edit operation of GED
- $\boldsymbol{c(.)}$**:** cost function of an elementary edit operation o_i
- $\boldsymbol{d(G^1, G^2)}$**:** computed GED, allowing to transform G^1 to G^2
- $\boldsymbol{\Gamma(G^1, G^2)}$**:** set of all edit operations $o = (o_1, \ldots, o_k$, allowing to transform G^1 to G^2
- $\boldsymbol{\epsilon}$**:** dummy vertex or edge which is used to model insertion or deletion operations
- $\hat{\boldsymbol{G}}$**:** maximum common sub-graph of G^1 and G^2
- $\check{\boldsymbol{G}}$**:** minimum common super-graph of G^1 and G^2
- $\boldsymbol{D^1}$**:** set of edit operations that are required to transform G^1 to $\hat{G}$
- $\boldsymbol{D^2}$**:** set of edit operations that are required to transform $\hat{G}$ to G^2
- $\boldsymbol{(x, y, u, v, e, f)}$**:** 6-tuple of binary variables which is used to define an edit path between the graphs G^1 and G^2
- $\boldsymbol{|G|}$**:** number of vertices (G_v) in the graph G
- $\boldsymbol{A^v}$**:** finite or infinite attribute or label set for G_v
- $\boldsymbol{A^e}$**:** finite or infinite attribute or label set for G_e
- $\boldsymbol{a^v}$**:** vertex attribute of the graph G
- $\boldsymbol{a^e}$**:** edge attribute of the graph G
- $\boldsymbol{G_\mu}$**:** vertex labeling function which associates the attribute or label a^v to a vertex G_v^i
- $\boldsymbol{G_\nu}$**:** edge labeling function which associates the attribute or label a^e to a vertex G_e^i
- $\boldsymbol{f_v}$**:** substitution cost function of the labels of the substituted vertices
- $\boldsymbol{f_e}$**:** substitution cost function of the labels of the substituted edges
- $\boldsymbol{g_v}$**:** insertion/deletion cost function of the labels of the inserted/deleted vertex
- $\boldsymbol{g_e}$**:** insertion/deletion cost function of the labels of the inserted/deleted edge
- $\boldsymbol{G_v^s}$**:** source vertex of the graph G
- $\boldsymbol{G_v^d}$**:** destination vertex of the graph G
- $\boldsymbol{N_{G_v^s}}$**:** number of pixels of the source vertex (G_v^s) of the graph G
- $\boldsymbol{N_{G_v^d}}$**:** number of pixels of the destination vertex (G_v^d) of the graph G
- $\boldsymbol{ED_{G_v^{s,d}}}$**:** Euclidean distance between the two graph vertices (G_v^s and G_v^d)
- $\boldsymbol{m_{ji}}$**:** spatial moment
- $\boldsymbol{\mu_{ji}}$**:** central moment

- $\boldsymbol{\nu_{ji}}$**:** central normalized moment
- $\boldsymbol{hu_k}$**:** Hu moment
- $\boldsymbol{(\bar{x}, \bar{y})}$**:** mass center
- $\boldsymbol{F_e^{s,d}}$**:** edge force
- $\boldsymbol{AD_e^{x(s,d)}}$**:** absolute difference between the two extracted region centroids (s and d) in the x-axis
- $\boldsymbol{AD_e^{y(s,d)}}$**:** absolute difference between the two extracted region centroids (s and d) in the y-axis
- $\boldsymbol{Th_e}$**:** edge threshold
- $\boldsymbol{N_{HRs}}$**:** number of the extracted homogeneous regions
- $\boldsymbol{\bar{W}_{cre}}$**:** creation weight of the vertex or edge in the built directed graph
- $\boldsymbol{\bar{W}_{sub}}$**:** substitution weight of the vertex or edge in the built directed graph
- $\boldsymbol{M^g}$**:** distance matrix obtained by computing the dissimilarity between the compared graphs
- $\boldsymbol{m_{i,j}^g}$**:** element of the distance matrix obtained by computing the dissimilarity between the compared graphs

Glossary

- **9D-SPA:** 9-direction spanning area algorithm

A

- **AGNES:** agglomerative nesting
- **AIC:** Akaike information criterion
- **ANR:** "*agence nationale de la recherche Française*"
- **ARG:** attributed relational graph
- **ARLSA:** adaptive run-length smearing algorithm

B

- **BIC:** Bayesian information criterion
- **BH2M:** Barcelona historical handwritten marriages database
- **BHMD:** Barcelona historical marriage database
- **BLP:** binary linear programming
- **BnF:** "*bibliothèque nationale de France*"
- **BoVW:** bag of visual words
- **BoW:** bag of words

C

- **CAT:** computer-assisted transcription
- **CBIR:** content-based image retrieval
- **CC:** connected component
- **CCA:** connected component analysis
- **CCC:** cubic clustering criterion
- **CCl:** consensus clustering
- **CDI:** contemporary document image
- **CDIA:** contemporary document image analysis
- **CESR:** "*centre d'études supérieures de la Renaissance*"
- **CIIR:** center for intelligent information retrieval
- **CLARA:** clustering large applications

- **CLST:** minimum square-error clustering
- **CPU:** central processing unit
- **CRF:** conditional random fields
- **CRLA:** constrained run-length algorithm

D

- **DAS:** international workshop on document analysis system
- **Db2:** wavelet transform using 2-tap Daubechies filter
- **DHB:** digitized historical book
- **DI:** document image
- **DIA:** document image analysis
- **DIANA:** divisive analysis clustering
- **DIC:** document image classification
- **DIGIDOC:** document image digitization with interactive description capability
- **DIL:** document image layout
- **DILA:** document image layout analysis
- **DIU:** document image understanding
- **DL:** digital library
- **DLA:** discriminative locality alignment
- **DMLP:** dynamic multi-layer perceptron
- **DRR:** international conference on document recognition and retrieval
- **DTW:** dynamic time warping

E

- **EM:** expectation-maximization algorithm
- **EPF:** enhanced position formalism
- **ERC:** European research council

F

- **FBIM:** feature-based interaction map
- **FCBF:** fast correlation-based filter
- **FCM:** fuzzy c-means clustering
- **FFN:** feed-forward network
- **FP5:** European fifth framework program for research
- **FP7:** European seventh framework program for research

G

- **GDOH:** Gabor dominant orientation histogram
- **GED:** graph edit distance
- **GEDI:** ground-truthing environment for document images
- **GF:** Gabor filter
- **GFD:** generic Fourier descriptor
- **GLCM:** gray-level co-occurrence matrix
- **GLNU:** gray-level non-uniformity
- **GLRLM:** gray-level run-length matrix
- **GMM:** Gaussian mixture models
- **GMRF:** Gaussian Markov random fields
- **GPGPU:** general-purpose processing on graphics processing units
- **GSDM:** gradient spatial dependency matrix
- **GUI:** graphical user interface

H

- **HAC:** hierarchical agglomerative clustering
- **HBR:** historical book recognition
- **HBR2013:** historical book recognition competition 2013
- **HD:** historical document
- **HDI:** historical document image
- **HDIA:** historical document image analysis
- **HDIAR:** historical document image analysis and recognition
- **HDIL:** historical document image layout
- **HDILA:** historical document image layout analysis
- **HDIU:** historical document image understanding
- **HGRE:** high gray-level emphasis
- **HIP:** international workshop on historical document imaging and processing
- **HMM:** hidden Markov models
- **HNLA:** historical newspaper layout analysis
- **HSV:** hue, saturation and value space

I

- **i2S:** innovative, imaging, solutions

- **IA:** image analysis
- **ICL:** integrated completed likelihood
- **ICDAR:** international conference on document analysis and recognition
- **ICFHR:** international conference on frontiers in handwriting recognition
- **ICPR:** international conference on pattern recognition
- **ILP:** integer linear programming
- **IMPACT:** improving access to text
- **IOWC:** Indian ocean world centre
- **IPA:** image patches analysis
- **ISRI:** information science research institute
- **IST:** information society technologies program
- **IT:** information technology
- **IUT:** institute of technology

K

- **kNN:** k nearest neighbor

L

- **L3i:** "*laboratoire informatique, image et interaction*"
- **LaBRI:** "*laboratoire Bordelais de recherche en informatique*"
- **LBP:** local binary patterns
- **LGRE:** low gray-level emphasis
- **LI:** "*laboratoire informatique*"
- **LITIS:** "*laboratoire d'informatique, du traitement de l'information et des systèmes*"
- **LRE:** long-run emphasis
- **LRHGE:** long-run high gray-level emphasis
- **LRLGE:** long-run low gray-level emphasis

M

- **MDA:** multi-linear discriminant analysis
- **MDL:** minimum description length
- **MDS:** multi-dimensional scaling
- **MLP:** multi-layer perceptron
- **MRF:** Markov random fields
- **MST:** minimum spanning tree

N

- **NN:** nearest neighbor
- **NNS:** nearest neighbor search algorithm
- **NSF:** national science foundation

O

- **OCR:** optical character recognition
- **OFR:** optical font recognition

P

- **PAM:** partitioning around medoids
- **PCA:** principle component analysis
- **PGA:** pairwise geometric attributes
- **PPCM:** percentage of correctly classified pixels measure

R

- **RGB:** red, green and blue color space
- **RLF:** relative location features
- **RLNU:** run-length non-uniformity
- **RLSA:** run-length smearing algorithm
- **RLSO:** run-length smoothing with OR
- **ROC:** receiver operating characteristic
- **RPC:** run percentage
- **RXYC:** recursive XY-CUT

S

- **SAGE:** "*systèmes avancés en génie électrique*"
- **SAR:** simultaneous auto-regressive model
- **SDIP:** sparse discriminative information preservation
- **SED:** squared Euclidean distance
- **SIFT:** scale-invariant feature transform
- **SIMD:** single instruction, multiple data
- **SOM:** self-organizing maps
- **SP:** steerable pyramid
- **SRE:** short-run emphasis
- **SRHGE:** short-run high gray-level emphasis

- **SRLGE:** short-run low gray-level emphasis
- **SVM:** support vector machine

T

- **TCS:** texture co-occurrence spectrum

U

- **UBP:** unique bit pattern matrix

V

- **VLP:** visual language processing

W

- **WED:** weighted Euclidean distance

Chapter 1.

Introduction

This chapter introduces the context, challenges and overview of this work, and the key contributions and organization of this dissertation.

Contents

Since the early 1990s, libraries and museums have conducted large digitization campaigns with cultural heritage documents and scientific resources for ensuring restoration and lasting preservation of historical collections and promoting worldwide accessibility to cultural patrimony which requires to be protected from further deterioration and damages caused by repetitive handling [7]. Due to the huge amount of numeric high quality reproductions induced by the rapid growth of digital libraries worldwide, many challenges and open issues have been raised and have already spawned novel approaches and rigorous techniques of mass management. These solutions are designed to optimize the accessibility and navigability of huge mass and ever-increasing amount of available document images (DIs) (*i.e.* an easier browsing). Providing a reliable document interpretation system and developing an efficient content-based image retrieval (CBIR) tool which are oriented to historical document images (HDIs), are the prime necessities that have been pointed out to tackle the issues of large amount of data.

Recently, raising interest to document image analysis (DIA) and historical DIA has been generated, since it helps to reach the objective of ensuring the indexing and retrieval of digitized resources and offering a structured access to large sets of cultural heritage documents [8]. Indeed, an important need has emerged which consists in designing a computer-aided characterization and categorization tool, able to index or group digitized historical book (DHB) pages according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the HDI content.

This dissertation presents a number of studies and methods that address these challenges. The context (*cf.* Section 1.1), challenges (*cf.* Section 1.2) and overview (*cf.* Section 1.3) of this work, and the key contributions (*cf.* Section 1.4) and organization (*cf.* Section 1.5) of this dissertation, are presented in the following.

1.1. Context of this work

My thesis work has been carried out with the support of the French national research agency (ANR)[1] and the collaboration of many French research laboratories, "*laboratoire informatique, image et interaction*" (L3i)-University of La Rochelle[2], "*laboratoire d'informatique, du traitement de l'information et des systèmes*" (LITIS)-University of Rouen[3], "*laboratoire Bordelais de recherche en informatique*" (LaBRI)-University of Bordeaux I[4] and "*laboratoire informatique*" (LI)-University of Tours[5], in partnership with the French national library "*bibliothèque nationale de France*" (BnF) [6] and two industry partners, Arkhênum[7] and innovative, imaging, solutions (i2S)[8]. We are working on a project named DIGIDOC (document image digitization with interactive description capability)[9].

The DIGIDOC project aims mainly to simplify and improve the archiving, processing, comparison and indexing of DHBs. Specifically, its goal is to develop tools for analyzing HDIs throughout the acquisition process, from scanning the document to knowledge representation and management of HDI content. Moreover, the ultimate goal of the DIGIDOC project is developing relevant ways of interacting with scanners by assisting the digitization operator to adjust automatically the best set of parameters (e.g. resolution, lightening, color calibration), detecting errors in the digitization process (e.g. blur, skewed, folded pages), providing an appropriate assistance for document indexing (e.g. by recognizing automatically page types or breaks in a sequence of pages), *etc.* Indeed, there

[1] `http://www.agence-nationale-recherche.fr/en/`
[2] `http://l3i.univ-larochelle.fr/`
[3] `http://litis.insa-rouen.fr/`
[4] `http://www.labri.fr/`
[5] `http://li.univ-tours.fr/`
[6] `http://www.bnf.fr/fr/acc/x.accueil.html`
[7] `http://www.arkhenum.fr/`
[8] `http://www.i2s.fr/`
[9] The DIGIDOC project is referenced under "ANR-10-CORD-0020".
For more details, `http://www.agence-nationale-recherche.fr/en/anr-funded-project/?tx_lwmsuivibilan_pi2[CODE]=ANR-10-CORD-0020`

is an absolute need to design "smart" digitizers which can limit manual intervention and perform easy and high quality digitization of DIs [9]. Therefore, to achieve better interaction with scanners, we need to design a computer-aided categorization tool, able to index or categorize DHB pages according to several criteria, mainly the layout structure, graphical properties or typographical characteristics of the HDI content.

In this work, we are interested in tackling the fundamental problem of the DHB content characterization and DHB page categorization, with the goal of optimizing the accessibility and navigability of huge mass of HDIs. We have to find an alternative to the contemporary DIA tools which are mainly based on *a priori* knowledge of the layout and content of DIs to segment and characterize the analyzed DIs. Beyond this point, based on strong *a priori* knowledge, the contemporary DIA approaches are not effective if they are extended to be applied to a broader range of complex and degraded DIs such as the HDIs. Thus, the key task in this work is to show that it is possible to ensure automatic and relevant characterization and categorization of DHB pages without manual inspection or *a priori* knowledge regarding DI layout and content and with taking into consideration the particularities of HDIs.

1.2. Challenges of this work

Supported by the fact that pages of the same book usually present strong similarities in the organization of the HDI information (*i.e.* layout) and in the graphical and typographical features (*i.e.* content) throughout the DHB pages under consideration, our goal is to propose an approach that is used on an entire book instead of processing each page individually, for the segmentation and analysis of DHB content, and characterization and categorization of DHB pages. The aimed approach should not require *a priori* knowledge of the layout, typographical parameters or graphical properties of the analyzed DHB pages. It can extract automatically low-level features for discriminating the different classes of the foreground layers, through the analysis of the similarity and repetition information which is deduced from many DHB pages. Then, we aim to determine a region or group of pixels which share similar properties or characteristics on the basis of which they are grouped. These characteristics may be based on the localization of pixels and their surroundings, color, intensity or texture. In this work, we will focus only on texture-based features.

Recently, the issues of DIA have been considered as texture segmentation and classification [6]. It is commonly agreed that texture analysis plays a fundamental role for historical DIA and understanding since it has been considered as a consistent choice for meeting the need to segment a page layout under significant degradation levels and different noise types. Kise [5] stated that the analysis of pages with constrained layouts (e.g. rectangular, Manhattan) and clean DIs has almost been solved while historical DIA is still an open problem due to their particularities (e.g. noise and degradation, presence of handwriting, overlapping layouts, great variability of the page layout). He also precised that the most relevant methods used to analyze pages with unconstrained layouts and overlapping layers, are based on signal properties of page components by investigating texture-based features and techniques. Hence, texture-based methods address the challenges of the existing state-of-the-art ones and those initially dedicated to contemporary DIs. Given that there are significant degradations and no hypothesis concerning the HDI layout, the graphical properties or typographical parameters of the analyzed HDI, such as the type of script or handwriting (e.g. machine-print or printed, hand-print or manuscript, cursive), font size and type, scanning resolution, DI size, language, alphabet, *etc.*, the use of texture analysis techniques for HDI has become an appropriate choice. In addition, it has been shown that texture-based approaches work effectively with no *a priori* knowledge about the layout, content, typography, font and graphic styles, scanning resolution, DI size, *etc.* It has also been shown that they have good performance even for handwritten text. The use of a texture-based approach has been shown to be effective with skewed and degraded images.

In order to ensure a distinction between different text fonts and various kinds of graphics, three

assumptions are made [10]. First, the textual regions in a digitized DI are considered as textured areas, while its non-text content is considered as regions with different textures. Secondly, text with a different font is distinguishable. Finally, different types of graphics can be also separated. Thus, in this work various aspects of texture features have been explored in HDIs to assist the analysis of their content by characterizing a HDI through a set of homogeneous regions. Therefore, a data-driven or bottom-up strategy of analysis has been adopted in this work which is based on low-level data mining of pixels (e.g. texture, position, shape, geometry). This strategy investigates the texture and topology-based pixel properties (the spatial distribution of gray-levels) to determine the homogeneous or similar content regions in the analyzed HDI.

- First, faced with a large diversity of texture-based methods, few questions arise. Which texture methods are firstly well suited for segmenting graphical regions from textual ones, discriminating text in a variety of situations of different fonts and scales and separating different types of graphics ? Then, which texture approaches represent a constructive compromise between the performance (*i.e.* segmentation quality) and computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality) ? It is well-known that the success or failure of texture-based segmentation method tightly depends on the type of the extracted and used texture features. Thus, an experimental evaluation and benchmarking of a number of commonly and widely used texture approaches have been firstly conducted on a large corpus of HDIs, to have satisfactory and clear answers to the above questions. This work has shown the effectiveness of the different texture analysis approaches in the field of historical DIA.

- Given that there is a wide variety of DHB layouts and contents, having significant degradation levels and different noise types, proposing an approach that does not require any *a priori* knowledge, to characterize automatically DHB pages, is not a straightforward task. However, based on the hypothesis that some similarities of HDI content type can be deduced from many book pages and based on the assumption that a DI content type can be repeated on many pages of the same book [11, 12], a framework that works effectively at the entire book scale, instead of processing each book page individually, is proposed in this work. The proposed framework ensures the pixel-based characterization of the content of an entire DHB. It is automatic and it can be adapted to all kinds of books. It is independent of DI layout, typeface, font size, orientation, DI size, digitizing resolution and intensity, *etc.* It is also robust in the case of different kinds and levels of noise and degradation present in HDIs. Moreover, it does not require any manual inspection or *a priori* knowledge regarding DI content and structure or layout.

- A raising interest is noticeable recently to the use of statistical and structural pattern recognition tools to retrieve objects and classify them [13]. In DIA, the statistical and structural approaches are broadly applied for DI representation [14, 15]. A DI is represented by a feature vector in a statistical approach, while in a structural one a data structure (e.g. graph, tree) is used to model objects and their relationships in a DI. Therefore, by combining several points related to texture and topology-based segmentation methods and structural representation approaches that have been reported separately in the literature particularly on synthetic, medical and natural images, a structural signature based on texture, for each DHB page is proposed in this work. The proposed DHB page signature is characterized with a set of extracted homogeneous or similar content regions defined by similar texture, shape and geometric attributes and their topology. It does not assume *a priori* knowledge regarding the layout and content of the analyzed DHB pages, and hence, it is applicable to a large variety of ancient books. It integrates varying low-level features (*i.e.* texture, shape, geometric and topological descriptors) characterizing the different HDI content components (*i.e.* different text fonts or graphic regions) on the one hand, and structural information describing the HDI layout on the other hand. This rich and holistic representation of the layout and content of

the analyzed DHB page can be adapted to the user preferences and specified criteria through the extracted varying levels of information (e.g. by selecting only the information characterizing the HDI layout and/or content or by retrieving any useful information available for a subsequent use). It provides a topological signature of DHB page according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the HDI content.

- Finally, using the obtained signatures which are modeled in the form of graphs, the similarities of DHB page structure or layout and/or content can be deduced. To categorize and group DHB pages with similar layout and/or content, the obtained graph-based DHB page signatures can be compared using a graph dissimilarity. Then, the evaluation of the proposed page signature has been carried out based on computing a distance matrix, whose elements represent the dissimilarity between the compared graphs. Indeed, the DHB pages can be compared by categorizing the designed signatures which model the layout and content of DHB pages. In fact, DHB pages with similar layout and/or content can be grouped.

1.3. Overview of this work

The work conducted in this thesis proposes an automatic and relevant characterization and categorization approach of DHB pages. The proposed approach is independent of the layout and content of the analyzed DHB pages (*i.e.* it does not assume *a priori* knowledge regarding DI content and structure), and hence, it is applicable to a large variety of DHBs. It is based on the use of texture and structural information to provide a rich and holistic description of the layout and content of the analyzed DHB pages. The categorization is based on the characterization of the digitized page content by analyzing varying low-level of information (*i.e.* texture, shape, geometric and topological descriptors). More precisely, the signature-based characterization approach consists of two main stages. The first stage consists in extracting homogeneous regions. Then, the second one is proposing a graph-based page signature which is based on the extracted homogeneous regions, reflecting its layout and content. This signature ensures the implementation of numerous applications for managing effectively a corpus or collections of books (e.g. information retrieval in digital libraries according to several criteria or page categorization). To illustrate the effectiveness of the proposed page signature, a detailed experimental evaluation has been conducted in this work for assessing two possible signature-based applications for DHB page categorization, unsupervised page classification and page stream segmentation.

1. The characterization approach of DHB pages is based on identifying the different DI content components or blocks to characterize the DI layout and content and to define a page representation for each digitized page. The identification of the different DI content components or blocks is processed by extracting a set of regions of homogeneous texture or similar groups of pixels sharing some visual characteristics with their topological relationships. This would help modeling the layout structure, separating text from non-text regions, partitioning or categorizing pre-localized text blocks into columns, headings, paragraphs, lines, words, notes (head-notes and foot-notes) and abstracts, *etc.* Our goal is to extract as automatically as possible varying low-level features that segment DHB pages or a collection of DHBs into spatially disjoint homogeneous regions or similar content regions, without formulating a hypothesis concerning the DI structure or layout (e.g. column layout), typographical parameters (e.g. font size and type) or graphical properties (e.g. presence of embellishments) of the DI.

2. By characterizing each DHB page with a set of regions of homogeneous texture with varying low-level features, a structural signature, is designed for each DHB page. The proposed page signature integrates varying low-level features characterizing the different DI content components or blocks (*i.e.* text or graphic regions) on the one hand, and structural information describing the DI structure or layout on the other hand. This rich and holistic representation

of the layout and content of the analyzed DHB page can be adapted to the user preferences and specified criteria through the extracted varying levels of information (by selecting only the information characterizing the HDI content and/or structure or by retrieving any useful information available for a subsequent use). The extracted varying low-level information corresponds to the extracted *(i)* texture features to characterize the DI typographical and graphical characteristics, *(ii)* shape, geometric and topological features to describe the shape and spatial relationships of the extracted components of DI contents and *(iii)* structural information to take into consideration the page layout or structure.

3. The proposed page signature allows the implementation of several applications for managing effectively a corpus or collections of DHBs. To name a few, we may underline the following applications based on the defined page signature in this work:
   - Recognizing the analyzed page type to ensure an automatic adjustment of the quality of the page scanning process with respect to the page signature and subsequent use (*i.e.* designing a "smart" or "intelligent" scanner),
   - Modeling a computer-aided categorization tool, able to index, compare or classify DHB pages or DHBs according to several criteria (e.g. HDI layout and/or content) or to retrieve pages which have particular layout and/or content (e.g. empty or cover DHB pages),
   - Identifying specific pages, such as the transition pages in a DHB (e.g. title pages of chapter) which require a particular indexing process, to generate automatically a table of contents/summary of the analyzed DHB (*cf.* Figure 1.1),
   - Retrieving pages in a DHB that match specific criteria defined by a user (e.g. pages having particular layout and/or content),
   - Detecting pages having scanning failure occurring during the digitization process (e.g. blur, skewed, folded pages), *etc.*

Figure 1.1.: Example of a table of contents/summary of an analyzed DHB to generate using the proposed DHB page signature.

Among the numerous possible applications of the proposed DHB page signature (e.g. structure the whole HDIs corpus, index, retrieve, compare or group HDIs), a thorough evaluation has been conducted in this work for assessing two possible signature-based applications:

a) Unsupervised DHB page classification to group or gather similar layout and/or content DHB pages,

b) DHB page stream segmentation to generate automatically a table of content/summary of the analyzed DHB.

This evaluation has been carried out based on:

- Computation of graph edit distances (GEDs) between the different graph-based DHB page signatures, that can be used to retrieve similar pages in a HDI database query tool,
- DHB page categorization by analyzing the computed GEDs between the different graph-based DHB page signatures.

The assessment of the other potential applications of the proposed DHB page signature, cited earlier, will be among our future prospects.

Thus, the four main tasks describing the two analyzed and evaluated applications of the proposed DHB page signature are (*cf.* Figure 1.3):

a) ***Extraction and analysis of descriptors per region***

b) ***Generation of a structural signature (graph-based) per DHB page***

c) ***Computation of GEDs between page signatures***

d) ***Categorization of page signatures***

- ***Unsupervised DHB page classification*** (*cf.* Figure 1.2(a))
- ***DHB page stream segmentation*** (*cf.* Figure 1.2(b))

1.4. Contributions of this dissertation

This section summarizes the contributions of this dissertation, whereas the detailed contributions (along with the experiments and evaluations necessary to assess their performance) are discussed in the rest of the chapters of this dissertation.

1.4.1. Contributions

The main contributions of this dissertation are summarized in the following.

1. The first contribution of this work is presenting an experimental evaluation and benchmarking of a number of commonly and widely used texture features which have been conducted on a large corpus of HDIs for the purpose of determining the performance of each texture-based feature set according to the DI content, *i.e.* segmenting graphical regions from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand. To provide a qualitative measure of which texture-based feature sets are most appropriate for this task, nine texture-based feature sets (Tamura, local binary patterns (LBP), gray-level run-length matrix (GLRLM), auto-correlation function, gray-level co-occurrence matrix (GLCM), Gabor filters (GFs), 3-level Haar wavelet transform (Haar), 3-level wavelet transform using 3-tap Daubechies filter (Db3) and 3-level wavelet transform using 4-tap Daubechies filter (Db4)) have been investigated and assessed on 1100 pages of historical documents by using a classical texture-based pixel-labeling scheme for comparing texture features. The results reported in this work provide a useful benchmark in terms of performance, texture vector dimensionality, memory requirements, processing time and complexity for current and future research efforts in historical DIA. This work has also shown

the effectiveness of the different texture analysis approaches in the field of historical DIA, without formulating a hypothesis concerning the HDI layout (e.g. column layout) or its content (e.g. font size and type).

2. Then, the second contribution lies in the automatic analysis of characteristics of DHB pages (regarding their layout and/or content) to find homogeneous regions (*i.e.* graphic and textual regions) by analyzing texture features on an entire DHB instead of processing each page individually, with no assumption concerning the DHB page structure or layout (e.g. column layout), typographical or graphical properties (e.g. font size and type) of the DHB pages. Our goal is to provide a rich and holistic description of the layout and content of the analyzed book pages. Thus, a pixel-based characterization framework of the content of an entire book is proposed in this work, that can be seen as a first step towards to ensuring a simplified and user-friendly navigation on historical collections and cultural patrimony. Even if the typographical or graphical features are not known in advance, the texture information (e.g. typographical and graphical properties) which is often repeated and recurrently present in many DHB pages, can be deduced by exploiting the regularities of the associated textures through the whole DHB pages. So, in a first step, a clustering of texture features which are extracted from a sub-sampling in the entire DHB aims at identifying the texture information that is present in DHB pages. The clustering method that is applied has the ability to determine automatically the number of clusters or homogeneous regions. This knowledge is then used in a second step to segment each DHB page individually. Thus, a pixel-labeling framework which automatically analyzes texture descriptors by involving a multi-resolution/multi-scale approach to label pixels sharing similar textural characteristics (*i.e.* typographical and graphical properties) is presented as the second contribution of this work. The proposed pixel-labeling framework has been evaluated on a large variety of DHBs and achieved interesting results.

3. The third contribution of this work is proposing a structural DHB page signature to characterize DHB page structure or layout and/or content. This structural signature is designed for each DHB page, based on the set of the extracted regions of homogeneous texture (representing different DI content components or blocks) with their topological relationships, using a complete directed attributed graph. Where the graph vertices correspond to the extracted regions of homogeneous texture, and a set of edges is built based on topological relationships connecting the different vertices. The characterization of the extracted regions of homogeneous texture is based on varying low-level features (*i.e.* texture, shape, geometric and topological descriptors).

4. Finally, the last contribution of this work consists in illustrating the potential of the proposed graph-based signature by evaluating two possible signature-based applications, unsupervised page classification and page stream segmentation for DHB page categorization. To categorize and group DHB pages with similar layout and/or content, the obtained graph-based DHB page signatures can be compared using a graph dissimilarity. In our experiments, we use an approximate GED. The GED is used to measure the (dis)similarity between the obtained graph-based DHB page signatures [13]. The GED deals with the computation of the minimum-cost sequence of the basic graph editing operations (e.g. substitution, deletion and insertion of vertices or edges) to transform a graph to another one. The GED has to be set up based on the costs of the elementary edit operations (substitution, deletion and insertion). These costs are functions of the label of vertices/edges. Then, the evaluation of the proposed page signature has been carried out based on computing a distance matrix, whose elements represent the dissimilarity between the compared graphs. Indeed, the DHB pages can be compared by categorizing the designed signatures which model the layout and content of DHB pages. In fact, DHB pages with similar layout and/or content can be grouped. In this regard, a simple integrated user-centered graphical user interface (GUI) tool is designed for

the identification of the transition or similar layout and/or content pages in the DHB under consideration according to the user requirements.

In order to test the performance of the different proposed approaches in this work, detailed experimental evaluations on a large variety of DHBs and HDIs has been carried out. The evaluation has shown promising results in both accuracy and robustness.

1.4.2. List of publications

This dissertation has led to the following communications:

1.4.2.1. Journal papers

1. **M. Mehri**, P. Héroux, P. Gomez-Krämer and R. Mullot, Texture Feature Benchmarking and Evaluation for Historical Document Image Analysis. Pattern Analysis and Machine Intelligence, IEEE, 2015 [submitted].

2. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, A Texture-based Pixel Labeling Approach for Historical Books. Pattern Analysis and Applications, Springer-Verlag, pages 1-40, 2015.

1.4.2.2. International conference papers

1. **M. Mehri**, P. Héroux, J. Lerouge, P. Gomez-Krämer and R. Mullot, A Structural Signature Based on Texture for Digitized Historical Book Page Categorization. *International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015 [accepted].

2. **M. Mehri**, P. Gomez-Krämer, P. Héroux, M. Coustaty, J. Lerouge and R. Mullot, A Bottom-up Method Using Texture Features and a Graph-based Representation for Lettrine Recognition and Classification. *International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015 [accepted].

3. **M. Mehri**, P. Héroux, N. Sliti, P. Gomez-Krämer, N. E. B. Amara and R. Mullot, Extraction of Homogeneous Regions in Historical Document Images. *In Proceedings of the* 10^{th} *International Conference on Computer Vision Theory and Applications (VISAPP)*, SciTePress, Berlin, Germany, 2015.

4. **M. Mehri**, N. Sliti, P. Héroux, P. Gomez-Krämer, N. E. B. Amara and R. Mullot, Use of SLIC superpixels for ancient document image enhancement and segmentation. *In Proceedings of the* 22^{nd} *Document Recognition and Retrieval (DRR), Part of the IS&T/SPIE 27th Annual Symposium on Electronic Imaging*, SPIE, San Francisco, CA, USA, 2015.

5. **M. Mehri**, M. Mhiri, P. Héroux, P. Gomez-Krämer, M. A. Mahjoub and R. Mullot, Performance Evaluation and Benchmarking of Six Texture-based Feature Sets for Segmenting Historical Documents. *In Proceedings of the* 22^{nd} *International Conference on Pattern Recognition (ICPR)*, IEEE, pages 2885-2890, Stockholm, Sweden, 2014.

6. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, A Pixel Labeling Framework for Comparing Texture Features: Application to Digitized Ancient Books. *In Proceedings of the* 3^{rd} *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, SciTePress, pages 553-560, Angers, France, 2014.

7. **M. Mehri**, P. Héroux, P. Gomez-Krämer, A. Boucher and R. Mullot, A Pixel Labeling Approach for Historical Digitized Books. *In Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, pages 817-821, Washington, DC, USA, 2013.

8. **M. Mehri**, P. Gomez-Krämer, P. Héroux and R. Mullot, Old document image segmentation using the autocorrelation function and multiresolution analysis. *In Proceedings of the* 20^{th} *Document Recognition and Retrieval (DRR), Part of the IS&T/SPIE 25th Annual Symposium on Electronic Imaging*, SPIE, San Francisco, CA, USA, 2013.

1.4.2.3. International workshop papers

1. **M. Mehri**, N. Nayef, P. Héroux, P. Gomez-Krämer and R. Mullot, A Learning Texture-based Method for Enhancement and Segmentation of Historical Document Images. 3^{rd} *International Workshop on Historical Document Imaging and Processing (HIP)*, Tunis, Tunisia, 2015 [submitted].

2. **M. Mehri**, V. C. Kieu, M. Mhiri, P. Héroux, P. Gomez-Krämer, M. A. Mahjoub and R. Mullot, Robustness Assessment of Texture Features for the Segmentation of Ancient Documents. *In Proceedings of the* 11^{th} *International workshop on Document Analysis System (DAS)*, IEEE, pages 293-297, Tours, France, 2014.

3. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, Texture Feature Evaluation for Segmentation of Historical Document Images. *In Proceedings of the* 2^{nd} *International Workshop on Historical Document Imaging and Processing (HIP)*, ACM, pages 102-109, Washington, DC, USA, 2013.

1.4.2.4. National conference papers

1. **M. Mehri**, M. Mhiri, P. Gomez-Krämer, P. Héroux, M. A. Mahjoub and R. Mullot, Étude comparative de trois ensembles de descripteurs de texture pour la segmentation de documents anciens. *In Proceedings of the* 8^{th} *"Colloque International Francophone sur l'Écrit et le Document" (CIFED)*, pages 41-56, Nancy, France, 2014.

2. **M. Mehri**, V. C. Kieu, M. Mhiri, P. Héroux, P. Gomez-Krämer, M. A. Mahjoub and R. Mullot, Évaluation de la robustesse des descripteurs de texture pour la segmentation d'images de documents anciens. *In Proceedings of the* 8^{th} *"Colloque International Francophone sur l'Écrit et le Document" (CIFED)*, pages 25-40, Nancy, France, 2014.

3. V. C. Kieu, **M. Mehri**, V. Rabeux, N. Journet and M. Visani, Génération d'images semi-synthétiques de documents anciens à des fins d'évaluation de performances et d'apprentissage. *In Proceedings of the* 8^{th} *"Colloque International Francophone sur l'Écrit et le Document" (CIFED)*, pages 199-214, Nancy, France, 2014.

1.4.2.5. National communications at scientific congresses without proceedings

1. **M. Mehri**, P. Héroux, P. Gomez-Krämer and R. Mullot, A structural method based on texture for ancient document image analysis. *ICDAR 2015 Doctoral Consortium*, Tunis, Tunisia, 2015 [accepted].

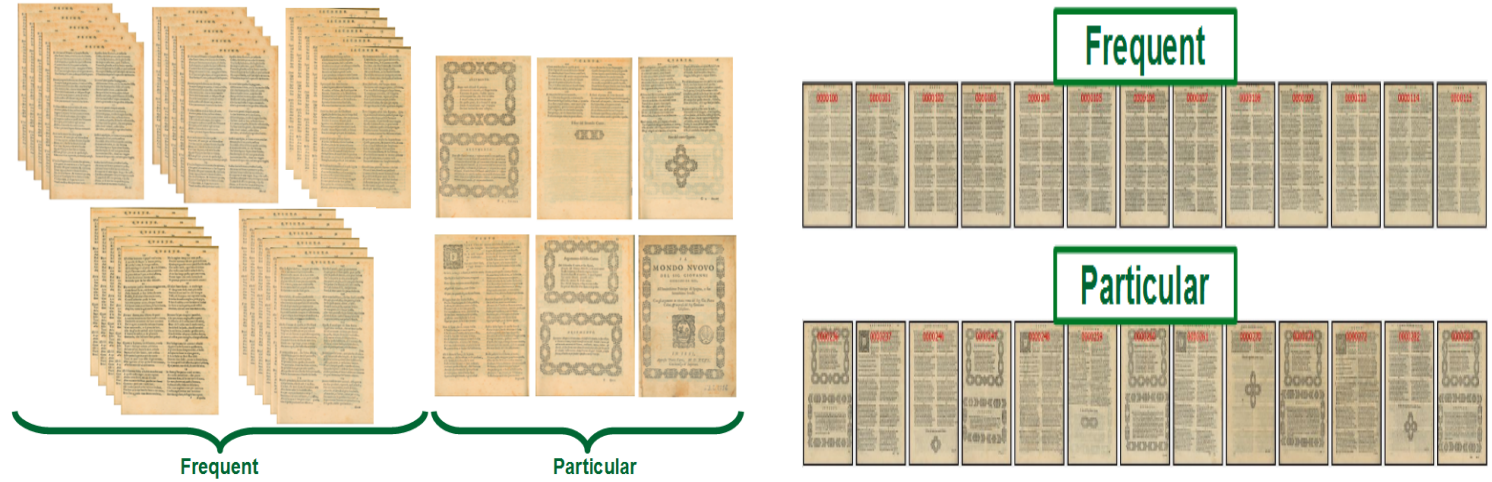
2. **M. Mehri**, P. Héroux, P. Gomez-Krämer and R. Mullot, Historical document image analysis: a structural approach based on texture. *Biennial Meeting of the French Research Group in Written Communication (GRCE)*, Paris, France, 2015.

3. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, Old document image segmentation using the autocorrelation function and multiresolution analysis. *Biennial Meeting of the French Research Group in Written Communication (GRCE)*, Paris, France, 2012.

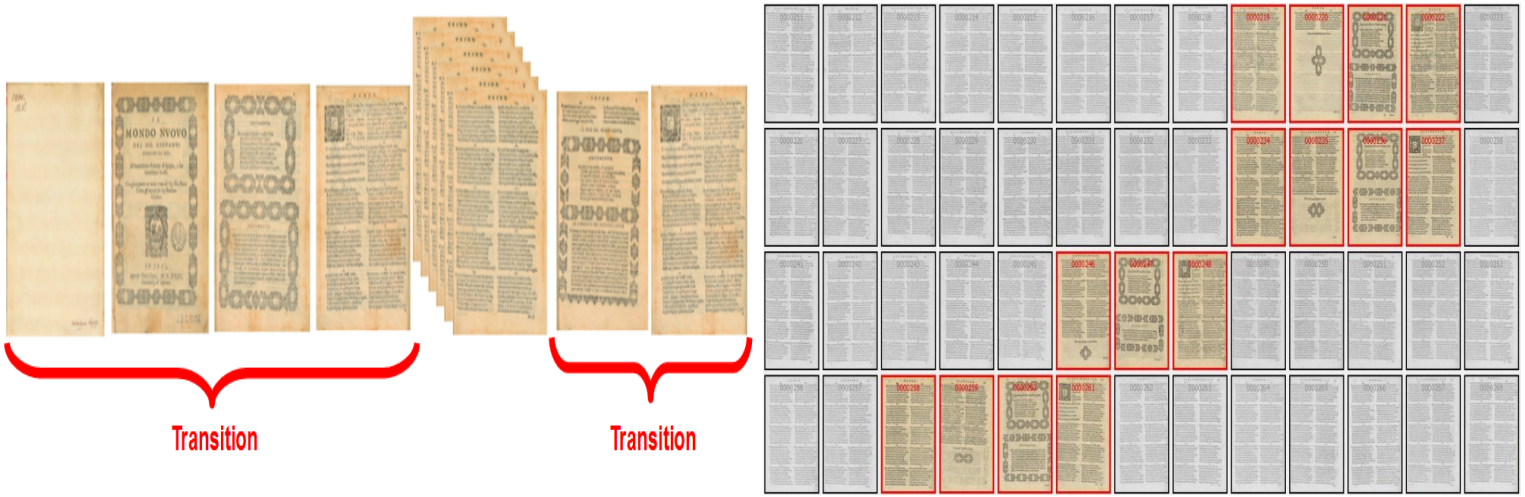
1.5. Organization of this dissertation

The rest of this dissertation is organized as six chapters:

- Chapter 2 reviews the research projects related to digital libraries and historical DIA.
- Chapter 3 outlines the related works on DIA and different texture-based methods proposed in the literature with a particular focus on those related to DIA and historical DIA.
- Chapter 4 presents an experimental evaluation and benchmarking of a number of commonly and widely used texture features which have been conducted on a large corpus of HDIs.
- Chapter 5 presents a texture-based pixel-labeling framework for DHBs.
- Chapter 6 presents a structural signature based on texture used for DHB page characterization.
- Chapter 7 presents two applications of the proposed signature for DHB page categorization in the context of DIGIDOC project.
- Chapter 8 summarizes some conclusions about the work presented in this dissertation and possible future directions of the work.



(a)



(b)

Figure 1.2.: Illustration of the two assessed applications of the proposed signature in this work: DHB page stream segmentation and unsupervised DHB page classification. Figure (a) illustrates the application of the proposed signature for unsupervised DHB page classification. DHB pages containing only text regions are illustrated on the first line, while DHB pages containing graphic and text regions are depicted in the second line. Figure (b) illustrates the application of the proposed signature for DHB page stream segmentation. By identifying the transition pages in a DHB which may correspond to the title pages of each chapter, a table of content/summary of the analyzed DHB is automatically generated.

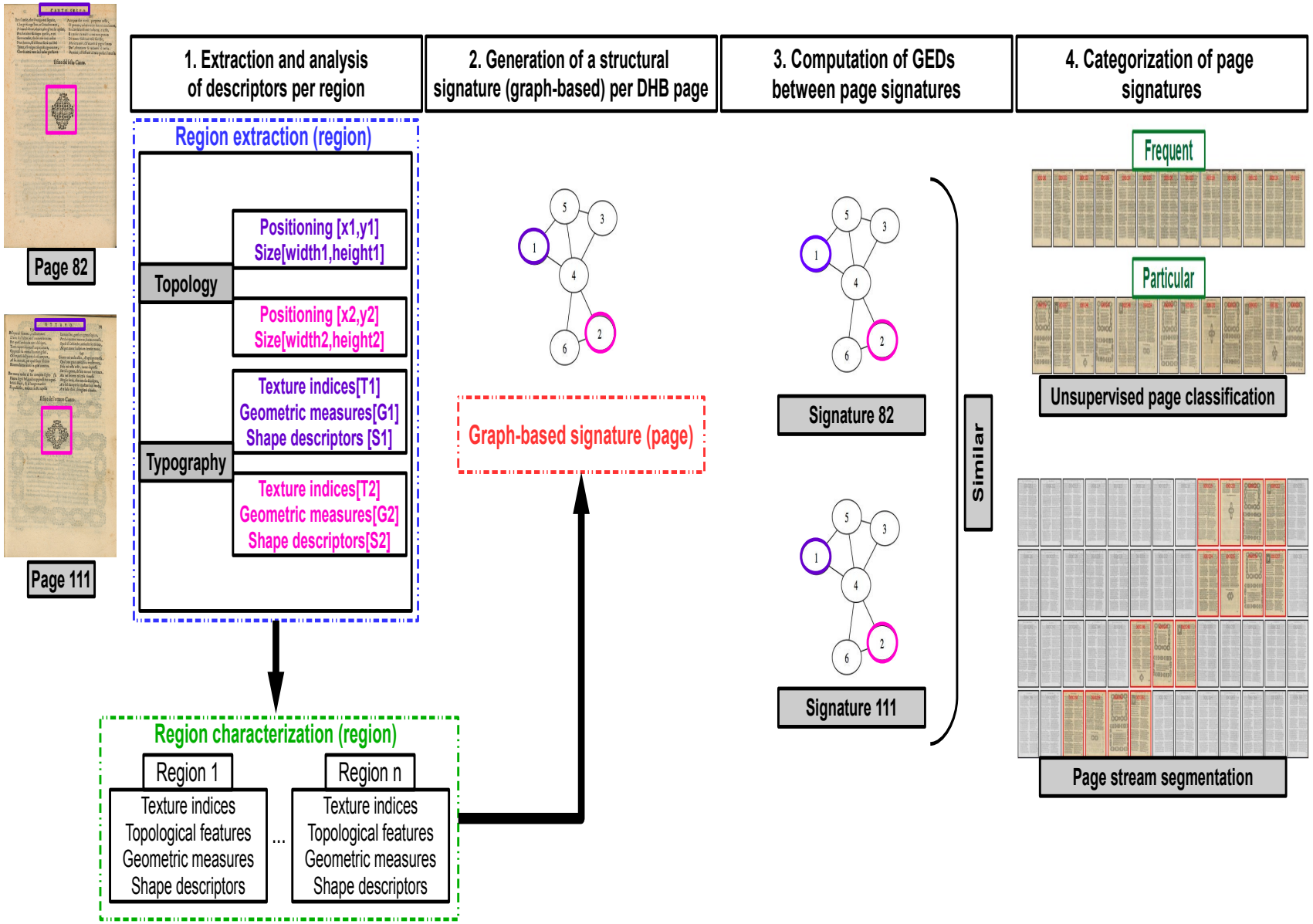


Figure 1.3.: Overview of the different steps of this work.

Chapter 2.

Digital libraries and challenges

This chapter reviews the research projects related to digital libraries and historical document image analysis. A number of initiatives have taken place to conduct large digitization programs with cultural heritage documents. Thus, new specific issues and challenges concerning the preservation and reproduction of historical collections have recently been addressed. The objectives and scope of this research work are given at the end of the chapter.

Contents

2.1. Introduction

The development of the Internet and electronic publishing, the prospects offered by the standardization of documentary techniques and broadcast media and the increased storage capacity and transmission rates, raise questions and pose specific challenges concerning the preservation and reproduction of historical collections. Thus, in order to guarantee a lasting preservation of historical collections and to provide a world-wide access to material which needs to be protected from too frequent handling, libraries have conducted large digitization programs with cultural heritage documents.

The idea of conducting strategies of digitization programs with cultural heritage documents has emerged since the early 1960s. The primary goals of these digitization programs which were related to the tremendous growth and spread of the Internet technologies, were not clearly identified (e.g. providing digital copies of historical documents, sharing databases of DIs between many libraries, designing a computer-assistance tool for textual data handling). Nevertheless, the significant digitization programs date to the 1970s, whose primary objectives were to preserve the historical collections and reproduce searchable, browseable and available on-line DI databases. From the 1990s onwards, new technologies have revolutionized the world of librarianship and printing thanks to the technological breakthrough and political decisions [16]. The European[1] and American[2] ministries of culture support digitization programs and encourage the development of digital libraries which offer new services such as on-line consulting of HDIs, fragile books and rare collections, *etc.* The French digital library Gallica[3], the British library[4] and the John F. Kennedy library[5] have been established for the purpose of preserving and exploiting this cultural heritage, and managing, promoting and developing digital supports of the cultural patrimony.

A number of initiatives have been taken to preserve and exploit the cultural heritage. For instance, the European library[6] is an on-line portal which provides quick, easy and open access to the collections of the 48 national libraries of Europe for research community world-wide. DELOS is a network of excellence on digital libraries, funded by the European commission[7]. Its primary objective consists in ensuring world-wide access to networked virtual libraries, by providing access to HDI collections residing in traditional libraries, museums, archives, universities, governmental agencies, specialized organizations and individuals around the world. In addition, it ensures the coordination and support of the efforts of the major European research teams working in digital library fields[8]. A complementary initiative of the DELOS activities which is called "DL.org - Digital Library Interoperability, Best Practices & Modelling Foundations" was set. Its primary goal is ensuring a focused approach to future great achievement in digital library related areas by forging strong links with the library and information science community, spanning educationalists, students, practitioners, researchers in book history, computer scientists, historians, librarians, end-users and decision makers[9].

At the industry sector level, numerous projects are in development to offer world-wide access to larger document collections and create global virtual libraries. The most well-known project is the "Google books library project" (previously known as the "Google print library project")[10] which has the objective to conduct a digitization and content indexing program of more than 15 million books of cultural heritage with the help of several libraries. It was initiated by the Google's partner program to offer the service of "Google books" (previously known as "Google book search"

[1] http://www.culture.gouv.fr/culture/mrt/numerisation/
[2] http://www.archives.gov/digitization/
[3] http://gallica.bnf.fr
[4] http://www.bl.uk
[5] http://www.jfklibrary.org/
[6] http://www.theeuropeanlibrary.org/tel4/
[7] http://ec.europa.eu/index_en.htm
[8] http://delos.info/
[9] http://www.dlorg.eu/
[10] https://www.google.com/googlebooks/library/

and "Google print"). This service provides an access to the full text of books and magazines that Google has scanned. The text in scanned DIs is automatically converted to editable text by optical character recognition (OCR) and stored afterwards in digital databases. This both ensures the access to the meaning of words in the pages, and easy and quick search for occurrences of words in the text.

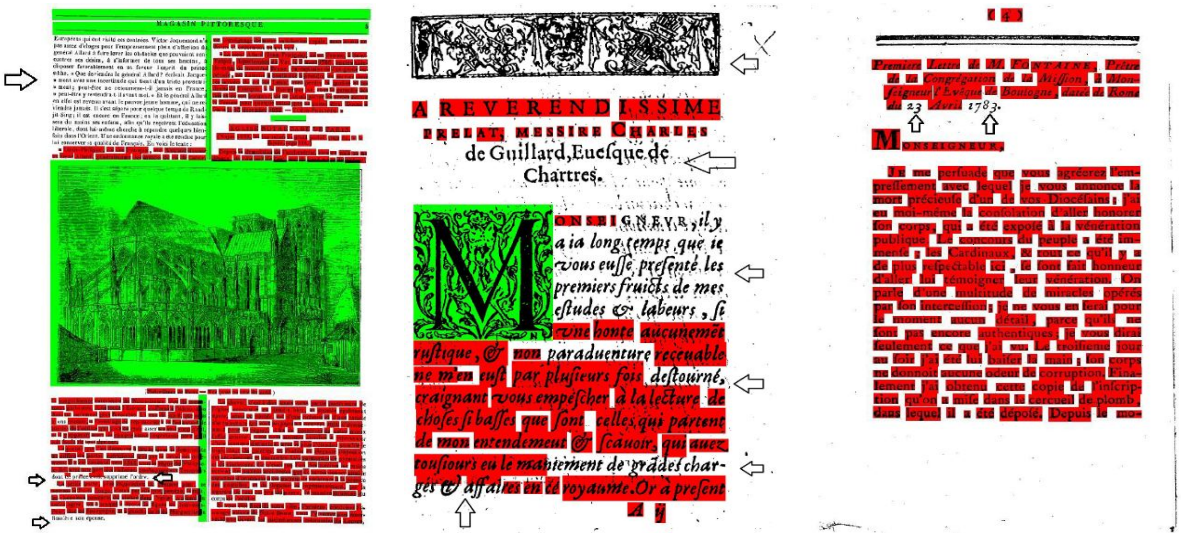
The increasing interest in digital libraries of Google and recently of other leaders and large firms (e.g. Microsoft, IBM, Yahoo, Amazon) proves the major success and effervescence of digital libraries and their rapid growth world-wide, and it poses new specific challenges concerning the preservation and reproduction of historical collections to reinforce its leadership position [17, 18, 19]. There has been an increase in special needs for information retrieval in digital libraries to optimize the exploitation of heritage documents [7, 20, 8]. As a matter of fact, this chapter introduces the challenges and goals of different research projects related to digital libraries and historical DIA, to meet the need to reinforce the enrichment and exploitation of heritage documents.

The remainder of this chapter is organized as follows: Section 2.2 presents a brief description of the main issues related to HDI indexing, with a particular focus on those related to OCR. Section 2.3 reviews the research projects related to digital libraries and HDI analysis. New specific issues and challenges concerning the preservation and reproduction of historical collections and the objectives and scope of this research work are presented in Section 2.4.

2.2. Towards historical document image indexing

One issue of particular concern is to provide a computer-based access and analysis of cultural heritage documents, searchable and browseable HDI databases, and an automatic indexing, linking and retrieval semantic-based systems of HDIs. As a consequence, there is a rapidly emerging need for an automatic conversion of text in digitized HDIs to editable text by the OCR. Since the early 2000's, investments in digitization must be accompanied by OCR to have access to full text content [21]. This ensures an automatic HDI indexing by textual content. In addition, this can also be useful in other contexts, such as the production of e-books, genealogy analysis, *etc.* For instance, the BnF has conducted many mass digitization projects in order to give access to its collection. The textual contents of the HDIs of the digital library of the BnF, Gallica, have been indexed by using OCR softwares since 2006 (*i.e.* textual transcriptions). OCR softwares have become more and more holistic, complex and sophisticated systems, and they do not only focus on solving a particular sub-task in restricted environments. They are composed of several modules dedicated to the analysis and recognition of the different page components. As a matter of fact, the OCR performance has been called into question for the OCR software inability to deal effectively with HDIs due to the HDI particularities (e.g. noise and degradation, presence of handwriting, overlapping layouts, great variability of page layout). Indeed, few errors such as the missed text components, can occur in OCR outputs due to the complexity of the OCR architecture and processing chain and the accumulation of errors at different levels in the OCR processing (*i.e.* pre-processing such as the binarization step, segmentation task or character/symbol recognition phase) [22, 23]. Few error examples of missed text/graphic components are illustrated in Figure 2.1. Therefore, manual corrections are required to ensure a high quality transcription. When the estimated word recognition rate is lower than 85%, manual corrections are recommended for the transcribed documents. Nevertheless, manual corrections are considered as expensive and exhausting tasks. In addition, the performance of the developed OCR softwares is highly dependent on the quality and particularities of the involved HDIs. And that is precisely why an automatic OCR verification phase should be integrated before the stage of HDI indexing by textual content. In fact, Salah *et al.* have recently proposed a texture-based approach for the detection of missed text components to control the OCR results from the Gallica collections [24].

The information science research institute (ISRI) which is a research and development unit of



(a) (b) (c)

Figure 2.1.: Illustration of the labeled masks detected by an OCR software [24]. Figure (a) shows an example of a missing section in the OCR output. Figure (b) depicts few error examples of missing words, sentences and graphical elements in the OCR output. Figure (c) illustrates few error instances of missing words in the OCR output. A red bounding box represents a recognized word by the OCR, while a green one represents a detected graphical element by OCR.

the University of Nevada[11], has conducted an annual "OCR technology assessment" program for benchmarking the OCR systems [25, 26, 27]. Its mission focuses on developing:

- New metrics of recognition performance,
- Measures of print quality,
- DI enhancement methods,
- Characterization of document analysis techniques.

The ISRI evaluated the OCR systems using a test data which is composed of five test samples (e.g. business letter samples, administrative document samples selected from the U.S. department of energy, magazine samples, English and Spanish newspaper samples). Few criteria were defined to evaluate the OCR systems under consideration (e.g. character accuracy, word accuracy, sentence accuracy, confidence score, accuracy character class, automatic textual block detection, document block labeling, accuracy document block labeling, document quality, resolution impact). The result of this benchmarking work consists in determining 280 degradation parameters. These parameters have been categorized as follows [22]:

- ***Imaging defects*** (e.g. heavy/light print, heavy and light print, stray marks, curved baselines),
- ***Similar symbols*** (e.g. similar vertical symbols, other similar symbols),
- ***Punctuation*** (e.g. commas, periods, quotation marks, special symbols),

[11] `http://www.expervision.com/testimonial-world-leading-and-champion-ocr/annual-test-of-ocr-accuracy-by-us-department-of-energy-doe-university-of-nevada-las-vegas-unlv`

- ***Typography*** (e.g. italics and spacing, underlining, shaded backgrounds, unusual typefaces, very large/small print).

In the context of the digitization programs developed to establish digital libraries, it is obviously necessary to take account of the quality of the original historical documents and all HDI processing stages, from the acquisition, the textual content transcription to the OCR verification, indexing and digital library integration steps. There are many factors that can affect OCR output quality:

- Characteristics of the digitized books or documents (e.g. textual content, typography, illustrations, presence of mathematical formulas) and edition (e.g. publisher, publication date), *etc.*
- Characteristics of paper, print and preservation quality, ink, inking, font size and type, *etc.*
- Characteristics of digitization (e.g. digitization quality, scanner type and parameters).

It is worth noting that the current efforts to build up a relevant OCR system are certainly planned to recognize typed text or words. According to [28], all DI components that are not purely textual ones, can disrupt the OCR processing, such as images, tables, mathematical and chemical formulas, numbers, hieroglyphics, handwritten annotations, graphics, *etc.* Moreover, depending on the composition of the text and textual structure (e.g. paragraph arrangement, font type and size, columns, text direction, text color), the OCR processing difficulty varies considerably. Other factors concerning the digitized ancient books or HDIs make the OCR processing task difficult and complex, such as languages and alphabets (e.g. accents, word length, number of languages, alphabets and scripts), references and quotations, punctuation, *etc.* On the other side, factors concerning the book edition and digitization properties (e.g. paper quality, printing defects, degradation, black borders of DIs, darker areas in the binding margins, flat scan by opening pair of pages, noise generated by the scanner roller and sensor, contrast/brightness level, curvature, compression, dynamic DI) have a major impact on the performance of the OCR and retrospective conversion tools.

As a consequence, a numerous research directions of studying the cultural heritage, various studies and different contributions achieved on distinct sub-fields and tasks related to the issues surrounding historical DIA (e.g. pre-processing, enhancement, restoration, graphics recognition, HDI layout analysis, HDI analysis and recognition, HDI understanding) in order to ease the effective functioning of an OCR software and improve its performance for HDI indexing.

2.3. Research projects dedicated to historical document image analysis

To meet the need to reinforce the enrichment and exploitation of heritage documents in addition to make it electronically available for access via the Internet, many research projects have been set up with the support of public funding provided by the European and American governments. The main goals of these projects are to provide a computer-based access and analysis of cultural heritage documents, searchable and browseable HDI databases and an automatic indexing, linking and retrieval semantic-based systems of HDIs. Some works have been proposed to deal with the whole HDIs or DHBs [1], while others have been focused on investigating and analyzing parts of HDIs such as the graphic images (e.g. illustrations, drop caps [29]) or text, styles/fonts, handwritten annotations [30].

Nevertheless, the rapid growth of digital libraries has become a serious hindrance to promote wide efficiency and effectiveness in the management of this cultural heritage resources (*i.e.* quick and relevant access to the information contained therein) due to the huge amount of digital high quality reproductions of fragile books and the large mass of digital copies of rare collections. Moreover, a lack of comprehensive and strategic management tools has become an obstacle to optimizing the exploitation of heritage documents [7, 20, 8]. In fact, finding reliable systems for the interpretation of HDIs has been a topic of major interest for many libraries and the prime issue

of research in the DIA community. There has been an increase in special needs for information retrieval in digital libraries and historical DIA. Numerous research projects (e.g. 5CofM[12], HisDoc[13], HisDoc2.0[14], IOW[15], MEMORIAL[16], DocExplore[17], Europeana[18], Europeana Newspapers[19], DEBORA[20], Philectre [31], BVH[21], BAMBI[22], MADONNE[23], NaviDoMass[24], DMOS[25], METAe[26], PlaIR[27], Bovary[28], Passe-Partout[29], GRAPHEM[30], Word spotting: indexing handwritten manuscripts[31], Culture, inheritance and creation[32]) deal with the digitization, enrichment and exploitation of European and American ancient heritage resources. A summary table of several research projects dedicated to historical DIA, describing briefly their target objectives, the tasks to carry out and the used datasets and showing their results, are presented in Table 2.2.

For instance, the European project DEBORA aims to develop networked libraries by improving accessibility to the 16^{th} century books of Italy, France and Portugal [9, 32]. One of the main interests of the MEMORIAL project is to develop a digital document workbench ensuring the creation of distributed virtual archives of printed HDIs from former Nazi concentration camp museums across Europe [33]. Europeana is a research project funded by the European comission[7] aiming to digitize historical newspapers to make them available for open access for research community world-wide. One of the aims of DocExplore is to construct a historical DIA framework which provides computer-based access and analysis of historical manuscripts. The aim of the HisDoc project is to design generic processing approaches and tools for historical manuscripts which are independent of the scripting language [34]. As part of the BAMBI project, a set of specific processing tasks have been developed for the recognition of handwritten scripts and the localization of textual information and drop caps (*i.e.* a drop cap is an ornamental letter that was widely used in books over time to represent the first letter at the beginning of a paragraph or a chapter) [29] and to determine document structure (columns, rows and paragraphs), particularly for medieval documents [35, 36, 37]. The aim of the MADONNE project is to develop a toolkit that can be used to index heritage documents and categorize book pages [38]. One of the main interests of the Philectre project is to explore and review the contribution of the computerized and electronic techniques, the computer technology and the image processing tools to the researchers in literary sciences, especially geneticists and medievalists. In this context, Lecolinet *et al.* [31] proposed an interactive system devoted to the visualization and the editing of hypermedia documents from literary material including DIs and structured text. This system integrates many DIA modules for manuscript transcription and structured textual representation of HDIs.

A limited number of standard public datasets of HDIs and their associated ground-truths is used in the context of different research projects to handle HDIs [39]. A number of HDI datasets are freely

[12] http://dag.cvc.uab.es/projects/five-centuries-of-marriages
[13] https://diuf.unifr.ch/main/hisdoc/
[14] https://diuf.unifr.ch/main/hisdoc/hisdoc2
[15] http://indianoceanworldcentre.com/Team_9
[16] http://www.primaresearch.org/projects/MEMORIAL
[17] http://www.docexplore.eu
[18] http://www.europeana.eu
[19] http://www.europeana-newspapers.eu/
[20] http://cordis.europa.eu/libraries/en/projects/debora.html
[21] http://www.bvh.univ-tours.fr/presentation_en.asp
[22] http://www.ilc.cnr.it/viewpage.php/sez=ricerca/id=97/vers=ing
[23] http://madonne.univ-lr.fr
[24] http://navidomass.univ-lr.fr
[25] http://www.irisa.fr/ra2001/imadoc/fonde_grammaires_mn.html
[26] http://meta-e.aib.uni-linz.ac.at/
[27] http://www.plair.org/doku.php
[28] http://www.bovary.fr/
[29] http://www3.unil.ch/BCUTodai/app/todaiGetIntro.do?uri=todaiInfo&page=todaiLogo.html
[30] http://liris.cnrs.fr/graphem/
[31] http://ciir.cs.umass.edu/irdemo/hw-demo/wordspot_retr.html
[32] http://cluster13.ens-lyon.fr/

available for historical handwriting recognition and word spotting (e.g. IAM-HistDB[33], George Washington[34] [35] [40], Parzival[36] [41, 42, 43], Saint Gall[37], RODRIGO[38] [44], ESPOSALLES[39] [45], Barcelona historical handwritten marriages (BH2M)[40] [46], Montesquieu's and Flaubert's manuscripts[41] [47, 48, 30], Vesalius's manuscripts[42] [49]).

The George Washington dataset[34] contains 20 pages from two writers. The Parzival dataset[36] is a multi-writer historical database and contains 47 pages. The RODRIGO dataset[38] is a single writer database and contains 853 pages. The ESPOSALLES database[39] contains two types of page parts from the marriage license book which was written between 1617 and 1619 by a single writer. The first part of pages, contains 1747 licenses on 173 pages. The second one which is called index, is composed of pages of 29 text pages. The BH2M dataset[40] consists of 174 images of manuscripts from the 17^{th} century. Those datasets are being used in the context of different research projects to deal with handwritten documents of inheritance by developing innovative techniques and proposing different approaches. In the context of historical graphical image analysis, the BCU Lausanne[43] which is a library at the University of Lausanne, proposed a dataset of 100 images of the ornaments of the 18^{th} century collected from old books [50]. This dataset ensure the comparison of the used printing equipment. In the context of NaviDoMass project, 4000 drop cap images from the 16^{th} and 17^{th} centuries [44] have been collected for graphical part indexing in historical heritage [51]. It is obviously necessary to note the unavailability or lack of a standard public large dataset of HDIs and its associated ground-truth. Moreover, most available datasets contain only handwritten HDIs.

Few datasets of HDIs used in the context of different research projects are summarized in Table 2.1.

Table 2.1.: Datasets dedicated to historical DIA.

Dataset	**Category**	**Number of pages**	**Project**	**Use cases**
George Washington[34] [52, 40]	-Two writers -18^{th} century -English language -Longhand script -Ink on paper	20	-HisDoc -HisDoc2.0 -Word spotting: indexing handwritten manuscripts	Handwritten historical document analysis and characterization
Parzival[36] [41, 42, 43]	-Three writers -13^{th} century -Medieval German language -Gothic script -Ink on parchment	47	-HisDoc -HisDoc2.0	-Handwritten historical document analysis and characterization -HDI layout analysis

[33] http://www.iam.unibe.ch/fki/databases/iam-historical-document-database/

[34] http://memory.loc.gov/ammem/gwhtml/gwhome.html

[35] http://www.iam.unibe.ch/fki/databases/iam-historical-document-database/washington-database

[36] http://www.iam.unibe.ch/fki/databases/iam-historical-document-database/parzival-database

[37] http://www.iam.unibe.ch/fki/databases/iam-historical-document-database/saint-gall-database

[38] https://www.prhlt.upv.es/page/projects/multimodal/idoc/rodrigo

[39] http://dag.cvc.uab.es/the-esposalles-database

[40] http://dag.cvc.uab.es/the-historical-marriages-database

[41] http://www.bovary.fr/folios_liste.php?type=f&id=4&mxm=0101030105&recueil=1&page=25&nb=24

[42] http://www.bvh.univ-tours.fr/Consult/index.asp?numfiche=56&url=/resrecherche.asp?ordre=titre-motclef=andre%20vesale-bvh=BVH-epistemon=Epistemon

[43] http://www.bcu-lausanne.ch/english-speaking/

[44] http://navidomass.univ-lr.fr/ressources.html

Table 2.1 – continued from previous page

Dataset	Category	Number of pages	Project	Use cases
Saint Gall[37] [53, 34, 54, 55, 56, 57, 4, 58]	-Single writer -9^{th} century -Latin language -Carolingian script -Ink on parchment	60	-HisDoc -HisDoc2.0	-Handwritten historical document analysis and characterization -HDI layout analysis
RODRIGO[38] [44]	Single writer	853	HisDoc	Handwritten historical document analysis and characterization
ESPOSALLES[39] [45]	Marriage license book which was written between 1617 and 1619 by a single writer	202	5CofM	-Handwritten historical document analysis and characterization -HDI layout analysis
BH2M[40] [46]	Manuscripts from the 17^{th} century	174	5CofM	-Handwritten historical document analysis and characterization -HDI layout analysis
Montesquieu's and Flaubert's manuscripts[41] [47, 48, 30]	Handwritten HDIs from the 18^{th} and 19^{th} centuries	500	-Culture, inheritance and creation -GRAPHEM -MADONNE -Bovary	Handwritten historical document analysis and characterization
Not mentioned [50]	Images of the ornaments of the 18^{th} century collected from old books from the BCU Lausanne[43]	100	Not mentioned	Graphical part indexing in historical heritage
Not mentioned[42] [51]	Drop cap images from the 16^{th} and 17^{th} centuries	4000	NaviDoMass	Graphical part indexing in historical heritage

Table 2.1 – continued from previous page

Dataset	Category	Number of pages	Project	Use cases
Vesalius's manuscripts[42] [49]	Rare DHBs	85	-BVH -MADONNE	-Graphical part indexing in historical heritage -HDI layout analysis -Historical collection modeling and representation
Not mentioned [59, 60]	Damaged military form pages of the 19^{th} century	88, 745	DMOS	HDI layout analysis
IAM-HistDB[33]	-Parzival[36] -Saint Gall[37] -George Washington[35]	-74 hand-written historical manuscript images -60 medieval manuscript pages -20 pages from the George Washington papers	-HisDoc -HisDoc2.0	-Handwritten historical document analysis and characterization -HDI layout analysis

These projects have addressed very specific issues in the field of historical DIA [20, 8]:

- Handwritten historical DIA and characterization,
- Graphical part indexing in historical heritage,
- HDI layout analysis,
- Historical collection modeling and representation.

To further illustrate the specific main research themes, issues and dedicated services to historical DIA, the following is an outline of a categorization of few European and American research projects according to their goals and contributions.

2.3.1. Handwritten historical document analysis and characterization

In addition to the search and index of historical handwritten collections [52] and the word spotting and retrieval for HDIs [40] which are still open issues, other challenging issues have been presented such as handwriting recognition analysis [54], writer characterization and identification [61], handwriting classification [30], *etc.*

2.3.1.1. Culture, inheritance and creation

"Culture, inheritance and creation" is a French project carried out in collaboration with literary partners. It has investigated a digitized corpus of handwritten HDIs from the 18^{th} and 19^{th} centuries (*cf.* Figure 2.2) [30]. The aim of this work is to identify the authors of some of these manuscripts and to characterize and group together manuscripts written by the same author. Ancient handwritten manuscripts of some famous French authors (Montesquieu and Flaubert) were evaluated since the particularities of handwritten HDIs have been covered (*i.e.* they contain multi-writer annotations or corrections and characterized by background noise and degradation such as background spots, delocalized folds, *etc.*).

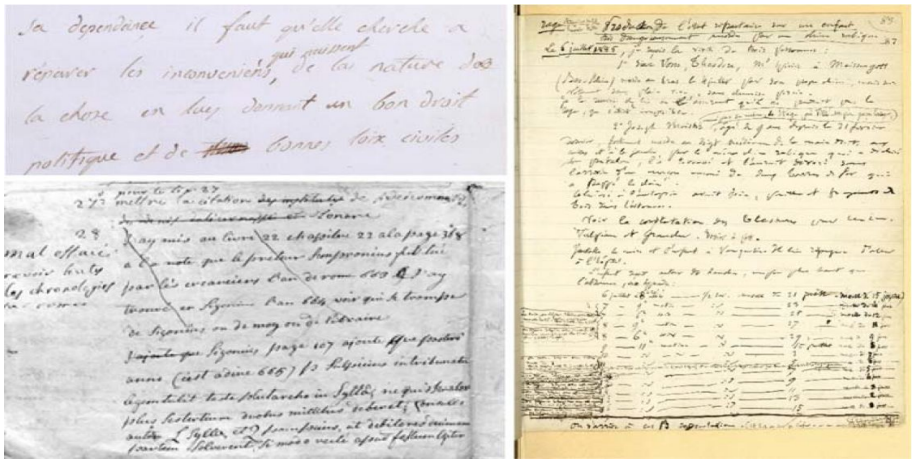


Figure 2.2.: Examples of ancient manuscripts collected from the French digital library Gallica[3]: Montesquieu's autograph "*De l'Esprit des Lois*" (1789) and Montesquieu's secretary (1780) [30].

2.3.1.2. GRAPHEM

GRAPHEM is another multi-disciplinary research project whose main goal is the automatic analysis of medieval writings to support palaeography experts in analyzing manuscripts which is a highly complex work that demands painstaking attention to detail. The project aims to firstly investigate and analyze the evolution of writing forms and secondly to develop efficient and automatic methods enabling accessing to manuscript contents based on word image similarity (*i.e.* word spotting and word retrieval). Eglin *et al.* [62] proposed several methods for handwritten content analysis, handwriting grapheme decomposition, grapheme analysis and classification *etc.* The developed algorithms can significantly help to transcript historical manuscripts and identify writing styles or writers [63]. Figure 2.3 illustrates segments of medieval manuscripts used in the GRAPHEM project.

2.3.1.3. Bovary

In the context of the MADONNE project and particularly the Bovary project[45], a French manuscript digitisation project dealing with Flaubert's manuscripts[41] (*cf.* Figure 2.4), Nicolas *et al.* [47, 48] proposed to enrich historical manuscripts by presenting an approach for segmenting and analyzing Flaubert's handwritten manuscript layout. They proposed a set of tools to help historians to characterize Flaubert's layout style and provide an on-line, structured access and browsing capabilities to an hyper-textual edition of "Madame Bovary" draft sets.

[45] `http://www.bovary.fr/`

Figure 2.3.: Examples of segments of medieval manuscripts used in the GRAPHEM project [62, 63].

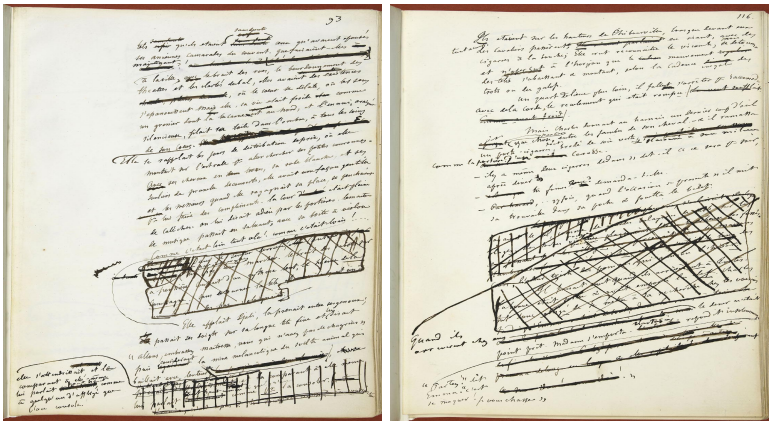


Figure 2.4.: Examples of Flaubert's manuscripts[41] collected in the context of the Bovary project [47, 48].

2.3.1.4. Word spotting: indexing handwritten manuscripts

A project on indexing handwritten historical manuscripts[46] has been developed by Rath *et al.* [52, 40] and supported by the center for intelligent information retrieval (CIIR) at the University of Massachusetts Amherst[47] and the national science foundation (NSF)[48]. They have used a part of the George Washington collection[34] at the library of Congress[49] (*cf.* Figure 2.5). Figure 2.6 shows two screen shots of the Web-based retrieval system interface[50] proposed by Rath *et al.* [52, 40] for handwritten text and line retrieval from the George Washington collection.

2.3.1.5. 5CofM

The 5CofM project is a Spanish research project. It is supported by the European research council advanced grant (ERC Advanced Grant)[51], and it is funded under the European seventh framework program for research (FP7)[52]. The main objective of 5CofM consists in extracting all the substantive information on five centuries of marriages contained in marriage license books (called Llibres d'Esposalles)[53], conserved at the archives of the cathedral of Barcelona, to produce a digital database which called the Barcelona historical marriage database (BHMD). The marriage register

[46] http://ciir.cs.umass.edu/irdemo/hw-demo/
[47] http://ciir.cs.umass.edu/
[48] http://www.nsf.gov/
[49] http://www.loc.gov/
[50] http://ciir.cs.umass.edu/irdemo/hw-demo/
[51] http://erc.europa.eu/advanced-grants
[52] http://ec.europa.eu/research/fp7/index_en.cfm
[53] http://dag.cvc.uab.es/5cofm-ground-truth

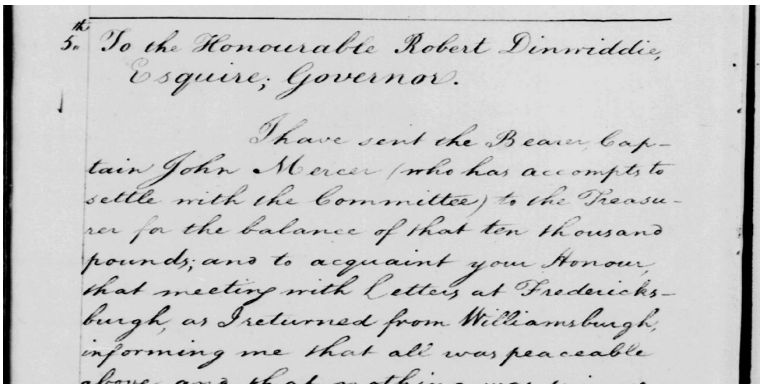
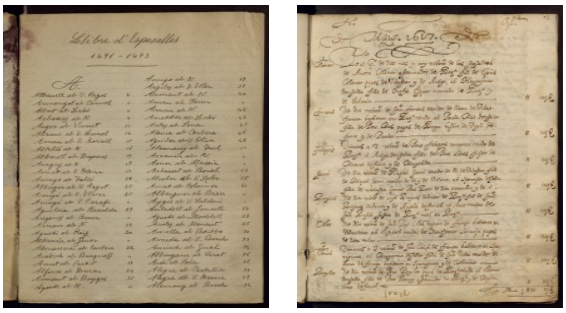


Figure 2.5.: Segment of digitized scanned manuscript document from the George Washington collection[34] [52, 40].

(a) (b)

Figure 2.6.: Screen shots of the Web-based retrieval system interface proposed by Rath *et al.* [52, 40] for handwritten text and line retrieval from the George Washington collection[34]. Figure (a) illustrates a screen shot of a ranked list of pages from 1000 handwritten page images in response to the "has" word, while Figure (b) shows a screen shot of the retrieved lines of handwritten HDIs in response to an input of a "has been" word query.

collection consists of 244 books containing approximately 600,000 unions celebrated between 1451 and 1905. It contains information concerning the recorded marriages and their corresponding fees paid according to the social status of the families. Two examples of marriage register collection pages can be seen in Figure 2.7. Figure 2.7(a) illustrates an index of marriage register collection book, while Figure 2.7(b) shows an instance of a marriage license. Two benchmarking databases (the ESPOSALLES and BH2M databases) are freely available to evaluate the most challenging tasks of a knowledge extraction and analysis process for automatic recognition and annotation of historical manuscripts such as word spotting [46] and HDI layout analysis [64, 65, 66], *etc.* The ESPOSALLES database[39] for handwriting recognition [45]. The BH2M database[40] for HDI layout analysis [64] and word spotting [67]. In the context of the 5CofM project, Fernández-Mota *et al.* [66, 46] proposed new approaches for handwritten text line segmentation and sequential word spotting in historical handwritten documents.



(a) Index (b) Marriage license

Figure 2.7.: Examples of marriage register collection pages from the Llibres d'Esposalles (archives of Barcelona cathedral)[53] [45, 46]. Figure (a) illustrates an example of an index of marriage register collection book, while Figure (b) shows an example of a marriage license.

2.3.2. Graphical part indexing in historical heritage

Most of the research projects addressed the issues related to text in HDIs for indexing and retrieval [1]. However, other studies examined historical graphical images such as the drop caps [29]. HDIs often contain graphical features represented by drop caps (*cf.* Figure 2.8). Drop caps were widely present in DHBs of the 15^{th} and 16^{th} centuries. They occur at the beginning of a chapter or paragraph. Other terms are often used to define a drop cap such as lettrine, drop capital or ornamental letter. A drop cap is usually represented by two main layers: a letter or initial and a drawing painted in the background. Analyzing drop caps ensures the indexing of the DHBs of the beginning of the printing period. Another interest of investigating and examining the drop caps in HDIs is to enrich semantically them by adding meta-data or semantic annotations. Thus, the drop caps can be described, classified and compared using the obtained signatures, and historical documents can subsequently be dated historically, authenticated or characterized by identifying differences between the analyzed drop caps. Other uses include developing relevant drop cap CBIR systems. The idea consists in providing a lettrine image query to the developed lettrine CBIR system, that will retrieve within a database all similar lettrines. There are several other needs expressed by the historian community. For instance, by analyzing drop caps the font, color or alphabet that characterizes the printer can also be deduced and investigated. In other cases, analyzing drop caps allows grouping the alphabet used in the drop caps, studying the wear of buffers used to print the drop caps, investigating the drawing painted in drop cap background and examining the progress of the used printing techniques, *etc.* The analysis of drop caps is considered complex, since there is a large variety and wide range of drop cap models, and they contain a lot of information (e.g. texture, letter, decorated background). In the context of the MADONNE and NaviDoMass projects, a number of studies have been carried out to extract specific parts from these complex graphic images and to compute signatures for indexing historical graphical images.

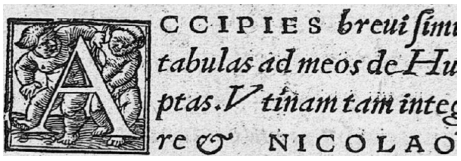


Figure 2.8.: Example of a drop cap [29].

Several research projects dealing with ancient illustration and ornament image datasets have been conducted [68]:

- The Fleuron project[54] has the objectives to provide a database of images of the ornaments to compare them and to investigate the pattern recurrence from one publisher to another, *etc.*
- The Môriane project[55] ensures the analysis of counterfeit ornaments used by printers of the 18^{th} century [69].
- The Passe-Partout project developed a software which is called TODAI[56], ensuring an automated search of a digitized ornament image from an ornament database by employing visual extraction and selection criteria without using textual description through keywords [70].

Nevertheless, the MADONNE, NaviDoMass and BVH projects appear to be the most recent, effective and successful research projects in the field of graphical part indexing.

2.3.2.1. MADONNE

The aim of the MADONNE project consists in developing a toolkit that can be used to index heritage documents and categorize book pages [38]. It is the result of fruitful cooperation between many French research laboratories between 2003 and 2006. Among the objectives of the MADONNE project is to develop a CBIR system for ancient graphical drop caps. Uttama *et al.* [29] examined drop caps from historical heritage images and introduced a drop cap segmentation method based on a combination of different texture features. A signature is afterwards assigned to a drop cap in order to design a CBIR system. A screen shot of the drop cap retrieval system interface proposed in the context of the MADONNE project is illustrated in Figure 2.9.

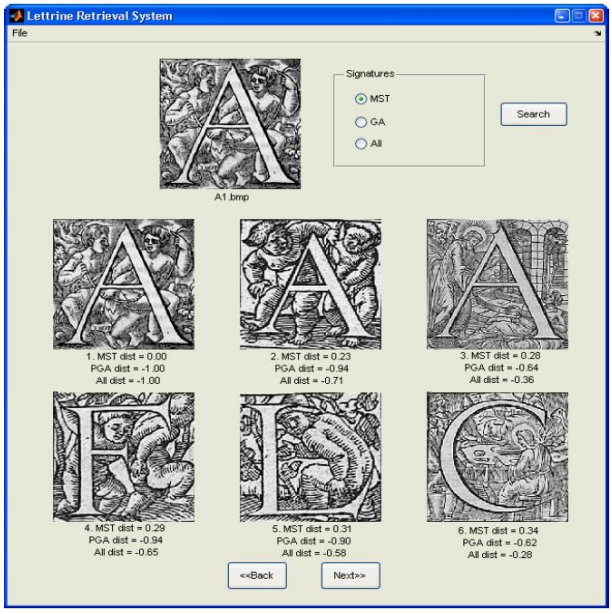


Figure 2.9.: Screen shot of the drop cap retrieval system interface proposed in the context of the MADONNE project [29].

2.3.2.2. NaviDoMass

The primary goal of the French research project NaviDoMass is to develop robust pattern recognition and analysis techniques supporting the particularities of HDIs (e.g. large variability of page

[54] https://apps.atilf.fr/fleuron2/
[55] http://www.enssib.fr/bibliotheque-numerique/notices/1510-le-projet-moriane
[56] http://www3.unil.ch/BCUTodai/app/Todai.do

layout, noise, degradation) ensuring rigorous description, classification and indexing of HDI collections by their content. In the context of the NaviDoMass project, Jouili *et al.* [15] proposed a structural-based framework to handle graphical images of HDIs (e.g. drop caps). They evaluated their approach on more than 4000 drop cap images[44] (*cf.* Figure 2.10), collected from the "*Centre d'Études Supérieures de la Renaissance*" (CESR)[57]. The CESR is a library, a training and research center. It ensures the access to a rich library of rare Renaissance books (*i.e.* from the 16^{th} and 17^{th} centuries) and support the efforts of the French research teams to work in various Renaissance-related areas. Coustaty *et al.* [51] proposed an approach for the extraction of drop cap letters by decomposing the information contained in the analyzed drop caps into several layers (*i.e.* segmenting the letter and the elements from its background) to characterize them by using a relevant signature.

Figure 2.10.: Examples of the drop caps collected from the CESR[57] [15].

2.3.2.3. BVH

The BVH project is a French project aiming to create a rich humanistic virtual library which provides a public Web-portal accessibility to more than 85 rare DHBs which were collected from the CESR, regardless barriers of time and place [49]. Providing a networked virtual library will enable anyone from their home, school or office to access the knowledge contained in the digital historical collections. In addition, the goal of the BVH project is to index these books in order to ensure new powerful technological capabilities that enable users to search among titles, authors, dates and other different queries relative to the digitized books to retrieve a particular book or book element (e.g. graphical or textual parts). Many kinds of graphics other than the drop caps, can be analyzed and retrieved. For instance, different ornaments (*cf.* Figure 2.11 [58]) and portraits (*cf.* Figure 2.12 [59]) were collected in the context of the BVH project to be analyzed. A Web-based retrieval system interface[60] of different kinds of graphics was proposed in the context of the BVH project[61]. An example of a search query of the medical illustrations in the Vesalius's manuscripts[42] in the Web-based retrieval system interface[62] is illustrated in Figure 2.14.

2.3.3. Historical document image layout analysis

HDI layout analysis consists in dividing a document page according to the nature of the extracted structure, such as separate text from non-text regions or partition text into columns, text blocks, lines, words, *etc.* It deals with the segmentation of a DI into homogeneous regions or zones which have similar properties to ensure coarse-level understanding of documents [71]. Specific challenges and open issues concerning HDI layout analysis have been posed and raised to deal with documents from the 15^{th}, 16^{th} and 17^{th} centuries in a number of research projects. By analyzing the layout or

[57] http://cesr.univ-tours.fr/

[58] http://www.bvh.univ-tours.fr/typographie.asp?offset=0

[59] http://www.bvh.univ-tours.fr/img_portrait.asp

[60] http://www.bvh.univ-tours.fr/Dionis/recherche_avancee.asp

[61] http://www.bvh.univ-tours.fr/Dionis/resultat.asp?auteur=VESALE%20Andr%E9&intraoper=Et&extraoper=Et&tri=Titre

[62] http://www.bvh.univ-tours.fr/Dionis/recherche_avancee.asp

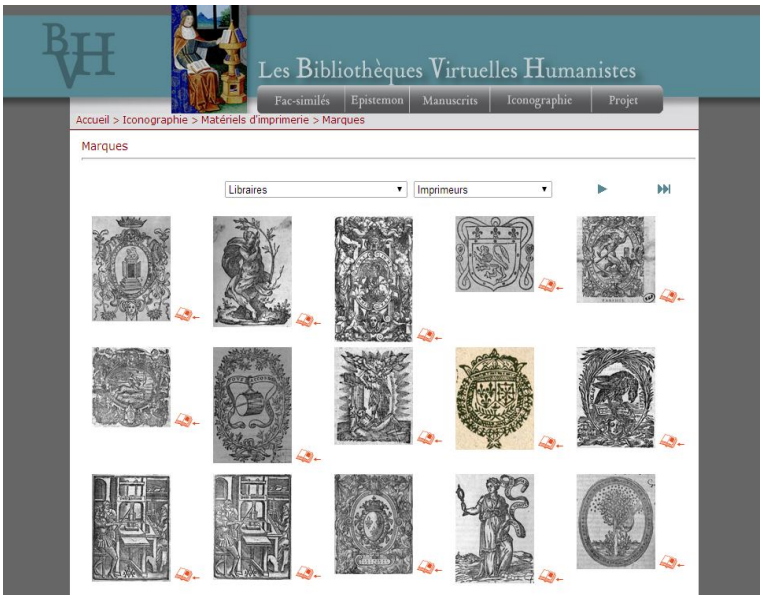


Figure 2.11.: Example of ornaments collected in the context of the BVH project[58].

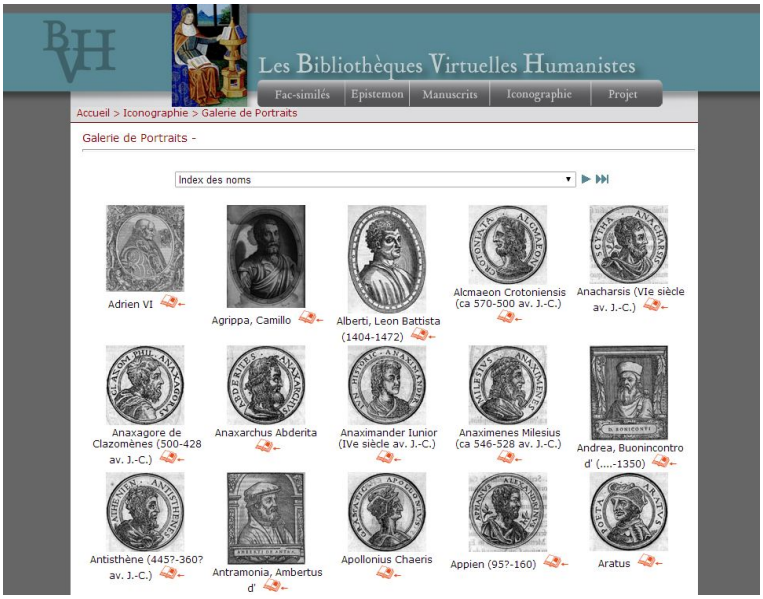


Figure 2.12.: Example of portraits collected in the context of the BVH project[59].

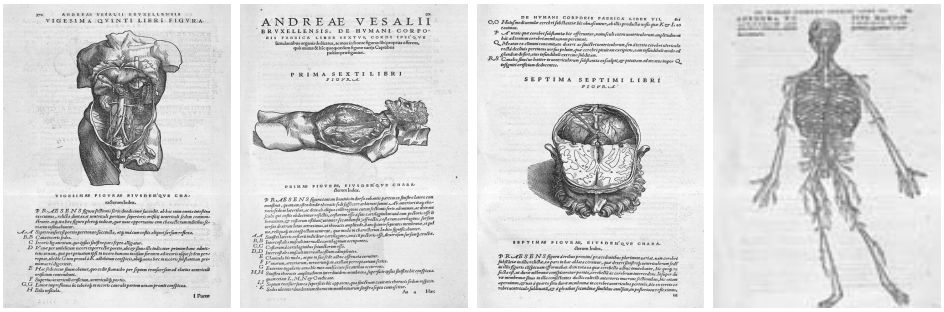


Figure 2.13.: Examples of different medical illustrations in the Vesalius's manuscripts collected in the context of the BVH project[61].

Figure 2.14.: Screen shot of the Web-based retrieval system interface of the medical illustrations in the Vesalius's manuscripts collected in the context of the BVH project[62].

structure of a DI, valuable information can be extracted and analyzed helping better description, understanding, browsing and indexing of the document content. Moreover, analyzing the DI layout or structure helps to investigate or examine the elements of layout, and subsequently it is possible to develop relevant CBIR systems capable to compare specific blocks yielded by the layout analysis and to design historical DIA and HDI recognition tools.

2.3.3.1. DEBORA

The DEBORA project is an European multi-disciplinary project which proposes a complete processing chain for retrospective conversion, analysis, indexing, retrieval and compression of digitized Renaissance books [32]. By extracting the meta-data related to the physical layout by means of connected component (CC) analysis technique and afterwards compressing images, the DEBORA project has been shown a suitable support for indexing, transmission, editing and annotation. The compression of book pages is based on an accurate segmentation of their content into different information layers, and it is adapted to the particularities of each extracted homogeneous region. It allows fast querying, navigation and downloading of required components of the logical structure and the physical layout of book pages. It is carried out by defining an appropriate electronic compressed file format adapted to the book page representation to improve access and browsing. In addition, to assist experts in manual transcription of Renaissance books, a computer-assisted transcription (CAT) was proposed in the context of the DEBORA project. The proposed CAT is able to transcribe all printed documents, regardless the used typography, language or alphabet [9]. A screen shot of the compressed file browser proposed in the context of the DEBORA project is illustrated in Figure 2.15.

2.3.3.2. BVH

In the context of the BVH project, an interactive HDI layout analysis and segmentation tool which is called AGORA[63], was developed to manage book content description. AGORA is a user-driven annotation tool which performs HDI layout analysis to index DHBs by extracting and structuring meta-data of indexing. In this context, Ramel *et al.* [72] evaluated various traditional methods used for segmentation of historical printed documents. They highlighted the limits of the traditional methods to segment HDIs. Thus, they proposed a hybrid segmentation algorithm based on CC analysis technique for the user-driven page layout analysis of historical printed books. The proposed algorithm used two maps, a shape map for foreground information analysis based on the CC analysis

[63] `http://www.rfai.li.univ-tours.fr/PagesPerso/jyramel/gb/work1.html`

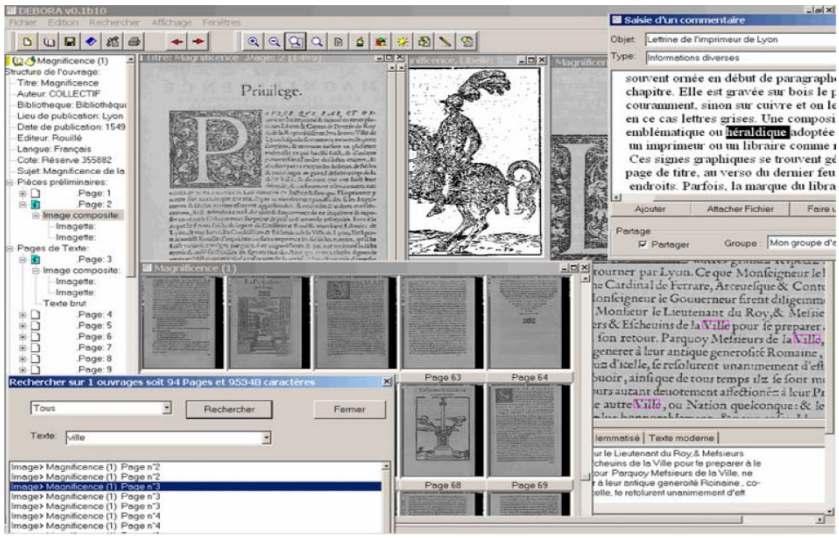


Figure 2.15.: Screen shot of the compressed file browser proposed in the context of the DEBORA project [32].

technique and a background map for white area analysis. Then, the classification of the extracted blocks by using the CC analysis technique was built according to scenarios defined by the user. With the use of simple descriptors (e.g. spatial position of the extracted blocks in the analyzed page, neighborhood relationships between the identified blocks, shape, block contents), the user can define many indexing scenarios corresponding to the selected book pages. Once the different indexing scenarios have been validated, they will be applied to the remaining of the book pages to index. An important use of the AGORA software is the automatic extraction and labeling of the graphical regions. For instance, when a user is interested in acquiring all drop caps in one or more ancient books to build an extensive database of drop caps, the following scenario can be defined (*cf.* Figure 2.16):

- The entity of the request is a drop cap,
- The entity query is always located in the 20% left of the image,
- The width/height ratio of the entity query is between 0.75 and 1.25,
- The closest right neighbor of the entity query is text.

Other uses of the AGORA software are, the extraction of the table of contents and the transcription of text blocks. Screen shots of the GUI of the AGORA software developed in the context of the BVH project are illustrated in Figures 2.16 and 2.17.

2.3.3.3. DMOS

A generic recognition method of 2-*D* structures was proposed in the context of the DMOS project. The proposed recognition method has been applied on many application domains (e.g. tennis court detection in videos, musical scores, mathematical formulas) among which HDI layout analysis [59, 60]. The HDIs processed by the DMOS project are very specific (old civil status registers and military forms of the 19^{th} century). They must have a strong, stable structure and especially describable by a set of rules defined by an expert user. In the context of the DMOS project, a software which is called FormuRead, was developed to extract automatically structure from quite damaged military forms of the 19^{th} century, found in French archives. FormuRead[64] was evaluated

[64] http://www.irisa.fr/intuidoc/index.php?option=com_content&view=article&id=72&Itemid=111&lang=en

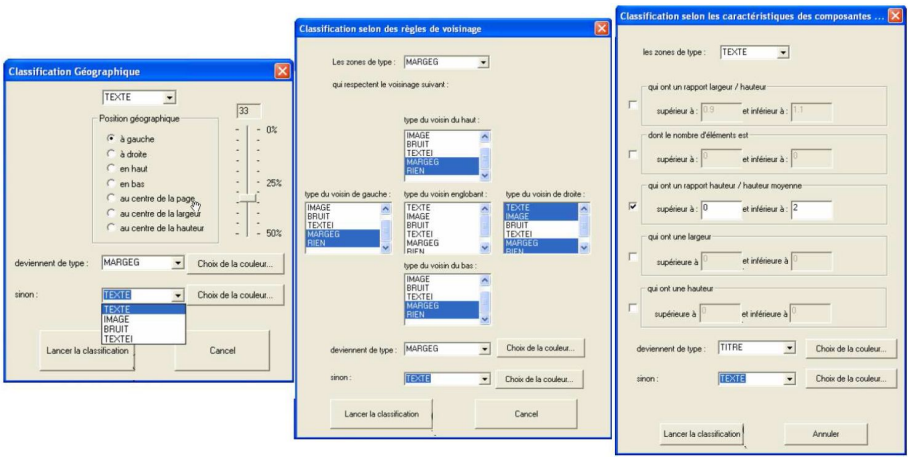


Figure 2.16.: Screen shot of the GUI of the AGORA software for the definition of indexing scenarios and the output result of the fusion of the CCs, developed in the context of the BVH project. An instance of a scenario to acquire all drop caps in one or more ancient books to build an extensive database of drop caps is illustrated by "the entity of the request is a drop cap, it is always located in the 20% left of the image, its width/height ratio is between 0.75 and 1.25, and its closest right neighbor is text"[63].

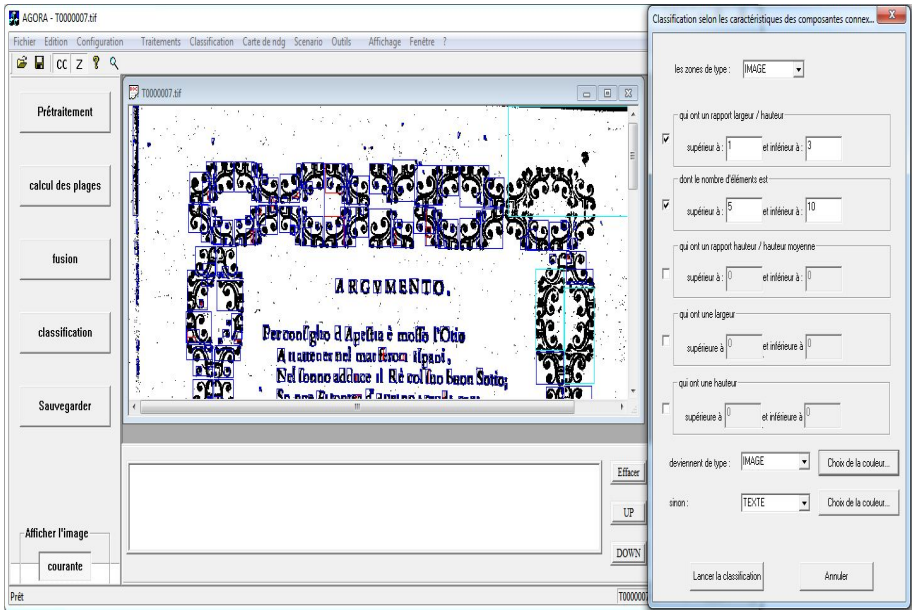
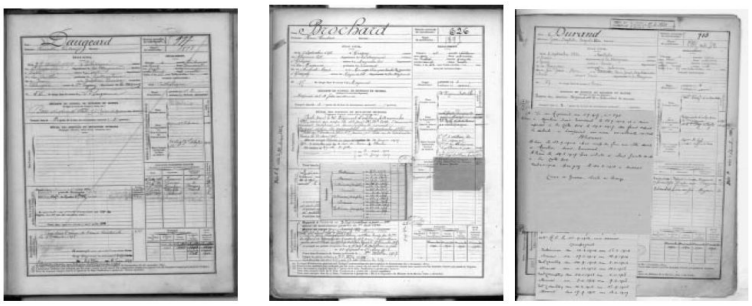


Figure 2.17.: Screen shot of the GUI of the AGORA software for the definition of indexing scenarios and the output result of the fusion of the CCs, developed in the context of the BVH project[63].

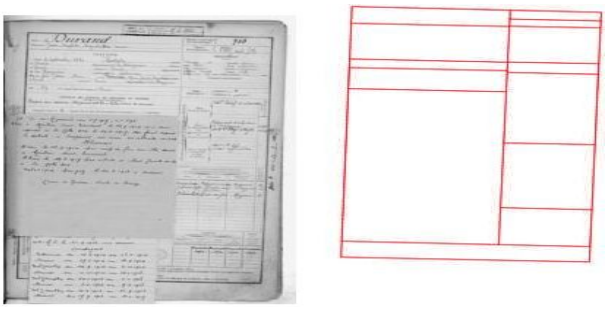
on 88,745 military form pages of the 19^{th} century. These military forms were collected from 140 registers of the archives of Mayenne[65] between 1878 and 1900 and 73 registers of the archives of Yvelines[66] between 1878 and 1885 (*cf.* Figure 2.18(a)). The evaluations have shown that the proposed recognition system has excellent structure extraction capabilities (*cf.* Figure 2.18(b)). Nevertheless, Coüasnon [59] stated that the military form analysis is an interesting example showing the difficulties and challenges for archive document recognition. Other interests which have followed

[65] `http://www.lamayenne.fr/fr/Archives53/Archives-en-ligne`
[66] `http://archives.yvelines.fr/article.php?laref=1`

the military form analysis have been pursued, such as making an automatic access to military form pages by handwritten content recognition (*i.e.* retrieve the right documents according to a textual request on the last name) after locating precisely the handwritten last name cell. Moreover, a DIA platform for managing all annotations was proposed to make handwritten archive documents accessible to public [73].



(a)



(b)

Figure 2.18.: Structure extraction of military form pages of the 19^{th} century with the FormuRead software which was developed in the context of the DMOS project. Figure (a) illustrates few examples of military form pages of the 19^{th} century. Figure (b) shows the result of the structure extraction of a military form page using the FormuRead software [59, 60].

2.3.3.4. METAe

The METAe project, in partnership with the BnF, was funded by the European Commission under the information society technologies (IST) program through the European fifth framework program (FP5) for research[67]. The idea behind the METAe project is to develop a set of tools able to digitize and analyze books (*cf.* Figure 2.19) and journals (*cf.* Figure 2.20) with a minimum of effort and a maximum of automation and effectiveness [74]. In this context, the German company, CCS[68] developed a software program known as DocWorks which offers an automated and structured conversion of printed ancient documents of the 19^{th} and 20^{th} centuries into digital formats. For easy access and searchability, DocWorks ensures the automatic recognition and description of the physical and logical document or book structure through the generation of image meta-data and character recognition using OCR. It can recognize specific fields (e.g. page numbers, titles, font sizes, page footnotes)

[67] `http://cordis.europa.eu/fp5/home.html`
[68] `http://content-conversion.com/?lang=en`

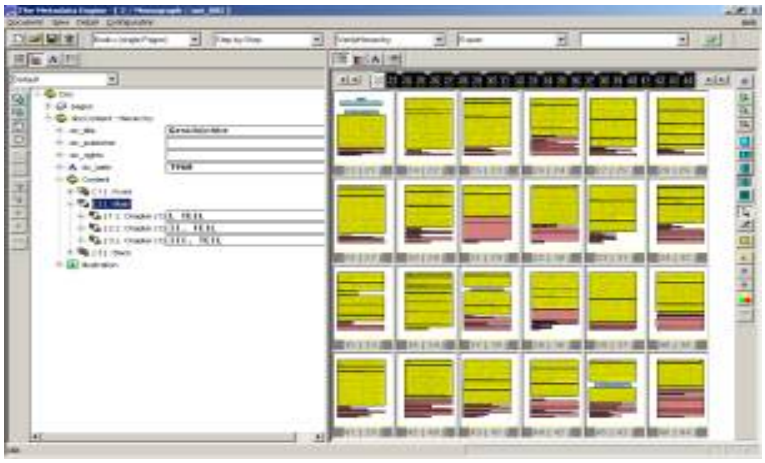


Figure 2.19.: Screen shot of the result of an automatic recognition of a book structure with the DocWorks software, developed in the context of the METAe project[68].

Figure 2.20.: Screen shot of the result of an automatic identification of different articles on a newspaper page with the DocWorks software, developed in the context of the METAe project[68].

2.3.3.5. PlaIR

PlaIR is a co-funded research project by the European union through the European regional development fund. The objective of the PlaIR project consists in developing a platform for indexing and searching of multi-domain and multi-purpose information from a set of digital library resources. By pooling a set of digital library resources and a number of software tools for automatic or semi-automatic analysis of these resources, the PlaIR project has been evaluated on the four following areas of application, health, engineering, law and scanned printed heritage archives. In the context of the scanned printed heritage archives, the main challenges are:

- Digitization of the archives of the "Journal of Rouen" newspapers from the years 1768 to 1848 (*cf.* Figure 2.21),
- Automatic identification of different articles on a newspaper page [75],
- Development of a efficient OCR engine based on the crowd sourcing technique, *etc.*

The PlaIR project proposes an on-line research and consultation application which is called PIVAJ, to offer a world-wide access to the digitized archives of the "Journal of Rouen" newspapers (*cf.* Figure 2.22) [76].

Figure 2.21.: Example of a newspaper page of the digitized archives of the "Journal of Rouen" newspapers used in the PlaIR project[27].

Figure 2.22.: Screen shot of the on-line research and consultation application which is called PIVAJ and developed in the context of the PlaIR project[27].

2.3.3.6. HisDoc

The HisDoc project is a Swiss research project dedicated to palaeographical analysis studies by proposing several methods for text localization, script discrimination and scribe identification in historical manuscripts. It has been supported by the Swiss national science foundation projects[69]. For historical manuscripts, a complete system for HDI layout analysis (*cf.* Figure 2.23(a)), handwritten text recognition (*cf.* Figure 2.23(b)) and information retrieval (*cf.* Figure 2.23(c)) was proposed in the context of the HisDoc project. The HisDoc research project is based on three distinct modules which are tightly linked[70] (*cf.* Figure 2.23) [34]:

1. ***Historical DIA***:
   The first module which is called historical DIA, involved two steps (*cf.* Figure 2.23(a)) [53, 56]:
   - HDI enhancement by modeling, understanding and eliminating the noise and degradation,
   - HDI layout analysis by describing and characterizing the layout and content of HDIs.

2. ***Handwritten text recognition***:
   The goal of the second module is to produce a fully automatic and robust segmentation and transcription system of text line images (*cf.* Figure 2.23(b)) [55, 54, 57, 4].

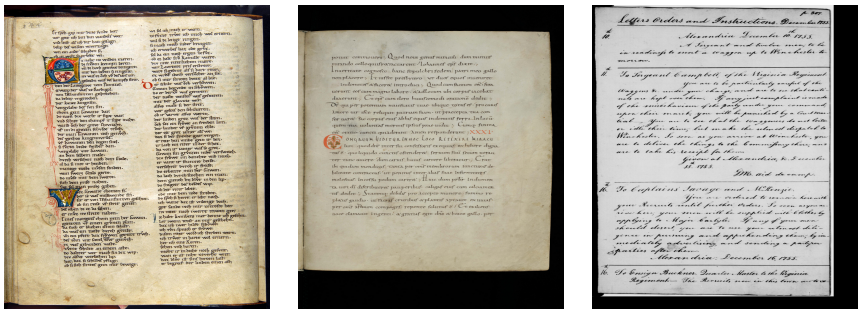
[69] `http://www.snf.ch/en/Pages/default.aspx`
[70] `https://diuf.unifr.ch/main/hisdoc/`

3. ***Information retrieval***:
   Finally, the third module has the main goal of implementing a search engine for noisy transcriptions provided by the second module of automatic handwritten text recognition (*cf.* Figure 2.23(c)) [77, 78].

(a) (b) (c)

Figure 2.23.: Illustration of the three complementary modules of the HisDoc project. Figures (a), (b) and (c) show the HDI layout analysis, handwritten text recognition and information retrieval modules, respectively[70] [34].

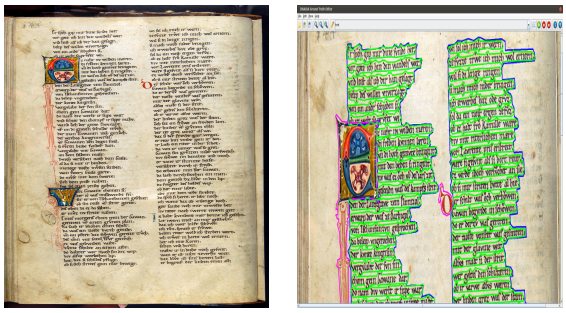
In the context of the HisDoc project, Fischer *et al.* [54] proposed an approach for automated reading of historical handwriting based on layout analysis and handwriting recognition modules. They have evaluated their system on the medieval Parzival database[36] (*cf.* Figure 2.25), and the proposed approach has achieved promising results [41, 42, 43]. The Parzival database which includes 47 pages, is a part of the IAM-HistDB[33]. The IAM-HistDB is a collection of datasets that contains handwritten historical manuscript images and is freely available. Two other datasets in the context of the HisDoc project have public access for evaluating handwriting recognition systems: the Saint Gall[37] and the George Washington[35] databases which consists of 60 medieval manuscript pages and 20 pages from the George Washington papers[34], respectively. The provided datasets in the context of the HisDoc project are all annotated, and the ground-truth contains both line-level and word-level transcriptions which were generated using the ground-truthing editor, known as DivaDia[71].



(a) (b) (c)

Figure 2.24.: Page examples of the three datasets freely available as parts of the IAM-HistDB in the context of the HisDoc project[33] Figures (a), (b) and (c) show three page examples of the medieval Parzival[36], Saint Gall[37] and George Washington[35] databases, respectively.

[71] `https://diuf.unifr.ch/main/hisdoc/divadia`



(a) (b)

Figure 2.25.: Evaluation of the automated reading of historical handwritings based on the layout analysis and handwriting recognition modules by means of the developed ground-truthing editor, known as DivaDia[71] in the context of the HisDoc project[33] [41, 42, 43]. Figures (a) and (b) depict an original DI collected from the medieval Parzival database[36] and its defined ground-truth, respectively.

2.3.3.7. 5CofM

In the context of the 5CofM project (*cf.* Section 2.3.1.5), the page segmentation step is an important task to retrieve textual information from huge data collections. Thus, Cruz-Fernández and Ramos-Terrades [64] proposed a document segmentation method based on relative location features (RLF) to segment structured HDIs collected from the 5CofM dataset[53]. This corpus is composed of highly structured pages. Each page contains a variable number of marriage license records and each marriage license record has three classes: the family name, record body and paid tax (*cf.* Figure 2.26). Cruz-Fernández and Ramos-Terrades [64] worked on segmenting these three classes. The experiment was performed on 512 pages of 5CofM database (volume 208) and they obtained good detection results of each of the three classes. In this same context, Álvaro *et al.* [65] defined a bi-dimensional extension of stochastic context-free grammars for page segmentation of structured documents. Moreover, Fernández-Mota *et al.* [66] proposed a graph-based approach for segmenting touching lines in historical handwritten documents. High performance was obtained comparing other state-of-the-art methods even the DIs contain skewed, multi-oriented, touching or overlapping lines. An illustration of the qualitative results of line segmentation obtained in the context of the 5CofM project is depicted in Figure 2.26.

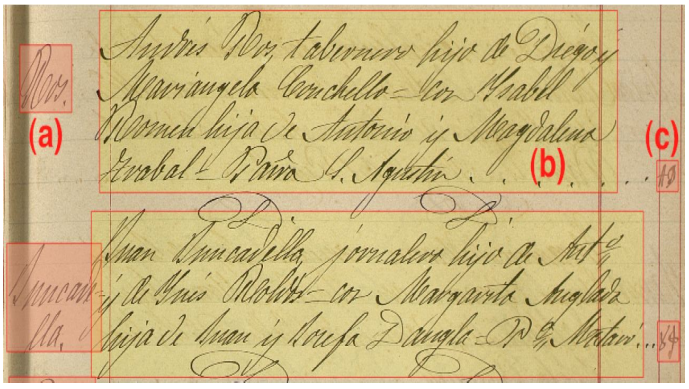


Figure 2.26.: Illustration of the defined ground-truth showing the document structure of 5CofM database (volume 208) for document segmentation used in the context of the 5CofM project [64].

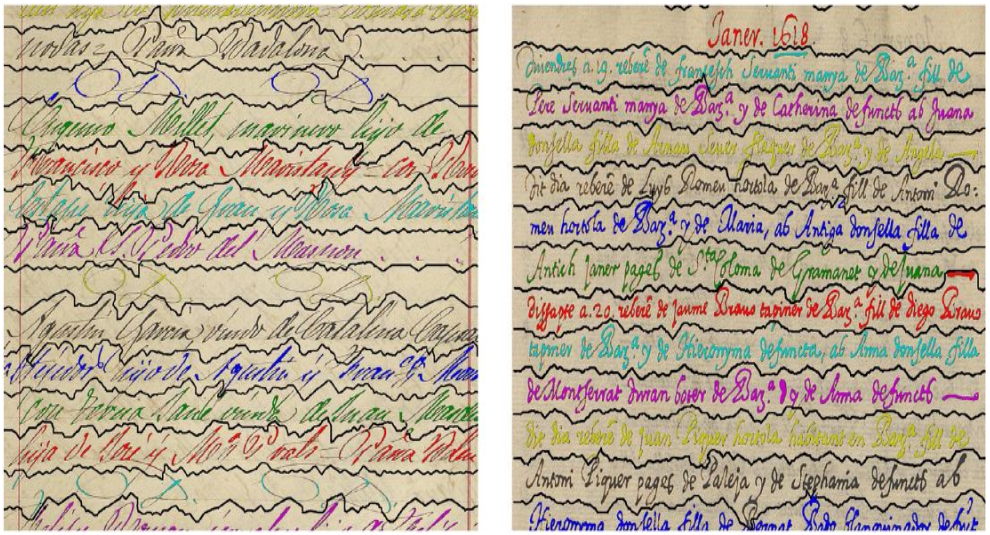


Figure 2.27.: Screen shot of the qualitative results of line segmentation obtained in the context of the 5CofM project [66].

2.3.4. Historical collection modeling and representation

It has been observed that some similarities of DI content type and a strong homogeneity of DI structure or layout can be easily deduced from many book pages or historical collections [11, 12]. This has led to a raising interest to provide innovative solutions related to extracting, modeling and representing knowledge in the context of large collections of data. Some encouraging efforts are noticeable in the context of the MADONNE and IOW research projects. The main goals of these initiatives is to identify the similarities concerning the collection structure by generating a relevant model summarizing each analyzed book and to categorize the book pages on the one hand and to determine the "social networks" linking historical collections and exploring the relationships among manuscripts and to characterize the interactions over the centuries and among the writers and cultures on the other hand.

2.3.4.1. MADONNE

In the context of the MADONNE project (*cf.* Section 2.3.2.1), Journet *et al.* [1] proposed an unsupervised texture-based approach for DHB content pixel-labeling. It was based on an unsupervised clustering technique using texture features which were extracted and analyzed from the pixel content of six pages of the same book by means of multi-scale approach. To assign the same label to pixels of six book pages which share similar textural characteristics, the clustering was performed on all extracted texture features of pixels of six book pages. The extracted texture descriptors were clustered and pixels were separated into two different content clusters, graphics and text. Then, the obtained pixel-labeling (graphics/text) was used as a basis for comparing the book pages and to group them into homogeneous classes. Figure 2.28 shows the interest of this simple approach by considering the number of pixels of drawing and text as a criterion to categorize or classify the book pages [79]. This classification is considered as the first step in a work of indexing HDIs, and subsequently it can help to characterize and identify the similarities concerning the collection structure (e.g. layout, typography) by generating a relevant model summarizing each analyzed book.

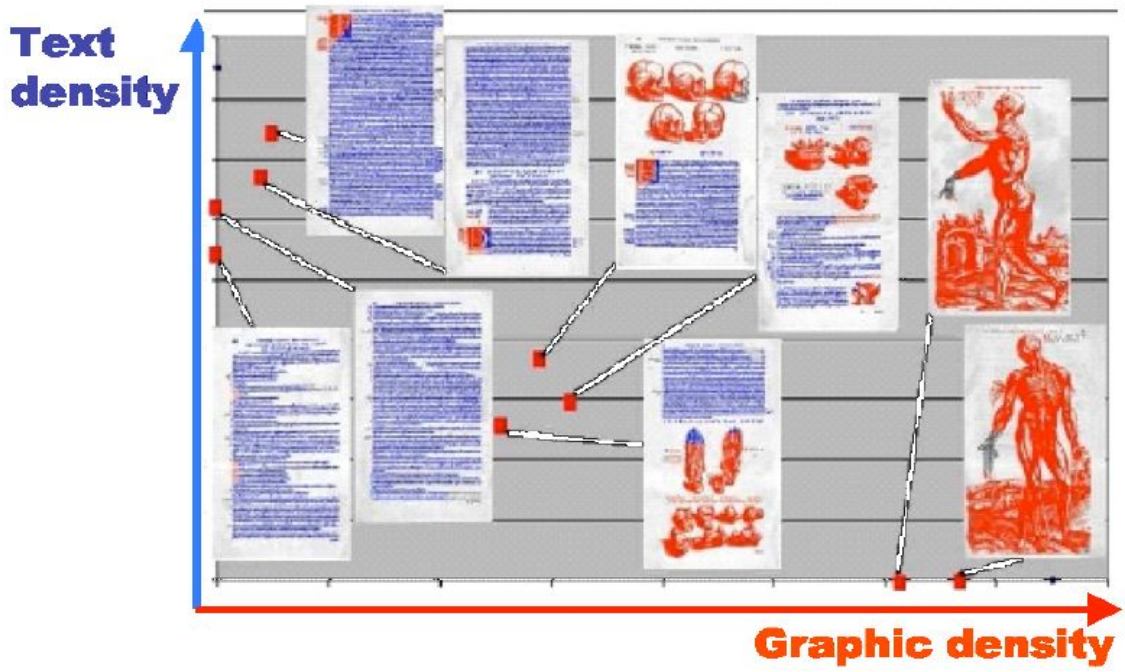


Figure 2.28.: Categorization of the book pages according to its content in the context of the MADONNE project [79].

2.3.4.2. IOW

The IOW project is an international and multi-disciplinary program of collaborative research which was funded by the social sciences and humanities research council of Canada (SSHRC)[72]. It aims to study and analyze the history of human-environment interaction in the Indian ocean world by exploring the relationships among manuscripts and identifying the interactions over the centuries and among the writers and cultures, after processing and understanding ancient manuscripts. The Synchromedia Lab[73] was involved in the IOW project for technical analysis of the collected documents (*i.e.* document enhancement, analysis and mining). Small collection of original manuscripts which was collected from the archives of the Indian ocean world centre (IOWC)[74] were digitized using multi-spectral imaging to acquire rich digital formats. Then, by automatically extracting data from the digitized documents, a dataset of multi-spectral HDIs was created, and text was transliterated and retrieved (*cf.* Figure 2.29).

In the context of the IOW project, Cheriet *et al.* [80] proposed a data-driven network-oriented analysis framework of historical manuscripts based on the visual language processing (VLP) for pattern analysis. The VLP is composed of three main levels:

- The lowest level focuses on automatic enhancement and restoration of ancient documents,
- The second one deals with the transliteration,
- The highest level copes with the analysis of the relationships between the extracted document components in the form of a network.

The VLP combines the visual class (images and manuscripts) and conceptual class (phrases and manuscripts) to determine and characterize the "social networks" linking ancient manuscripts (from the low-level relations of patches, excerpts, *etc.* to the high-level relations of manuscripts, collections, writers, *etc.*).

[72] http://www.sshrc-crsh.gc.ca/home-accueil-eng.aspx
[73] http://www.synchromedia.ca/node/336
[74] http://indianoceanworldcentre.com/archives

(a) (b)

Figure 2.29.: Illustrations of transliteration and transcription of historical manuscripts in the context of the IOW project[15]. Figure (a) illustrates the result of historical manuscript enhancement. Figure (b) shows the result of an automatic transliteration and transcription chain of historical manuscripts.

2.4. Achievements and open issues

Different research directions of studying the cultural heritage, various studies and different contributions achieved on distinct sub-fields and tasks related to the issues surrounding historical DIA including, pre-processing, enhancement, restoration, character recognition, graphics recognition, HDI layout analysis, HDI analysis and recognition, HDI understanding, *etc.* have been proposed in many specialized conferences and workshops (e.g. international conference on document analysis and recognition (ICDAR), international workshop on document analysis system (DAS), international conference on document recognition and retrieval (DRR), international workshop on historical document imaging and processing (HIP), international conference on frontiers in handwriting recognition (ICFHR), international conference on pattern recognition (ICPR)), as well in contests (e.g. historical document layout analysis, historical newspaper layout analysis (HNLA) and historical book recognition (HBR)) and journals (e.g. IJDAR, PR, PRL, PAA, PAMI, SMC).

However, Cheriet *et al.* [80] and Ogier [20, 8] stated that there is neither a generic method nor a unique solution to address all the issues and questions relating to processing HDIs. Cheriet *et al.* [80] highlighted the need to evaluate the proposed solutions on a large and varied amount of HDIs to prove their generality (*i.e.* to avoid bias introduced when performing the assessment on a small corpus). It is worth noting that by combining various algorithms and proposing multi-level and multi-stage frameworks, the exploitation of heritage documents will be optimized. Moreover, the high need of automation of DIA fields fulfills the requirements of optimization and supports the large international digitization programs with cultural heritage documents [74]. There has been an increase in special needs for information retrieval in digital libraries and HDI layout analysis [7, 20, 8]. Furthermore, a lack of comprehensive and strategic management tools has become an obstacle to optimizing the exploitation of heritage documents and to addressing the huge amount of HDIs. Indeed, many issues are still persistent and remain open such as [32]:

- High size of HDI files for the storage of HDIs,
- Lack of a standard file format exchange suitable for transmission,
- Lack of a common or standalone file format suited to HDI or DHB description and representation,

- Lack of relevant data compression techniques for faster remote access,
- Limited querying possibilities, *etc.*

Nevertheless, Ogier [20, 8] categorized the main open issues related to digital libraries and historical DIA into five classes:

1. ***Content characterization***:
   The issues related to the content characterization can be mainly linked to find the relevant signature for the indexing process. The defined signature is tightly dependent on the information to characterize the user requirements, subsequent use, *etc.* These signatures have been defined to deal with layout, handwritten or graphic indexing issues:
   - ***Layout-based***:
     For layout characterization, spatial signatures are defined after a segmentation stage to discriminate between the different classes of the foreground and background layers of a digitized document. For instance, Qureshi *et al.* [81] proposed an approach for symbol spotting using a graph representation of graphical documents (*cf.* Figure 2.30).
   - ***Handwritten***:
     A set of approaches have been developed for handwritten script recognition, authentication, transcription and textual information localization in handwritten HDIs. Those approaches defined some signatures by combining textural and/or spatial features. For instance, Wang *et al.* [82] proposed a coarse-to-fine word spotting approach for historical handwritten documents based on graph embedding and graph edit distance (GED) (*cf.* Figure 2.31).
   - ***Graphic***:
     A number of content-based characterization approaches have been developed in the context of graphic indexing. Those approaches are based on extracting several image and structural features. The image features are extracted from the analyzed image as a signature computed on the entire image and based on color, texture and/or shape. The structural features are deduced from a physical or logical layout analysis. Jouili *et al.* [83] proposed a structural representation in the form of a graph for historical graphical DIs. Based on the resulting graph-based representations, they assessed their approach to categorize drop caps (*cf.* Figure 2.32).
2. ***Scale resistance***:
   The need to evaluate the proposed solutions in historical DIA research fields on a large and varied amount of HDIs to prove their generality and efficiency (*i.e.* to avoid bias introduced when performing the assessment on a small corpus and when using a limited number of samples in the learning database) and to demonstrate their scale resistance due to the large variability of representations which is considered as one of the particularities of the HDIs. The idea consists in using relevant and generic features which better meet the needs of users for representing the class of objects to be indexed or recognized. For instance, Salmon *et al.* [84] combined shape descriptors based on a behavior study in order to improve the recognition rate of drop caps extracted from archival documents.
3. ***Management of large mass of digital rare collections***:
   By using a statistical and/or structural description to represent and characterize an object, a signature can be generated. This signature can help to retrieve an object or a part of an object in a database. In the context of dealing with a large mass of digital rare collections, it is not trivial to perform an exhaustive and sequential comparison of the query with all objects in the database due to the high computational complexity requirements. Thus, many studies have been proposed to structure the feature space and/or graph space by using various strategies depending on the type of the description used in the signature generation step. For

instance, Tabbone and Zuwala [85] proposed a combined filtering and indexing mechanism that retrieves in an efficient way the most similar symbols in graphical documents to a given input query. It is also important to emphasize that a raising interest is noticeable recently to the unification of structural and statistical pattern recognition tools to retrieve objects and classify them [13, 14]. In the context of the NaviDoMass project, Jouili *et al.* [15] proposed a structural-based framework for the drop cap clustering based on a graph-matching task. They proposed a structural representation for drop caps and compared the proposed approach with a statistical representation based on the generic Fourier descriptor (GFD).

4. ***User interaction***:
   Extracting a detailed description of the HDI content suited to the users' needs has become a major issue. Thus, a raising interest to the development of GUIs has been generated in order to respond swiftly to the users' requirements. Using these developed GUIs, the users can interactively define historical DIA scenarios according to their needs and expectations by proposing simple and editable scenarios. For instance, AGORA is a user-driven interactive annotation tool which performs HDI layout analysis to index DHBs by extracting and structuring meta-data of indexing [72]. Furthermore, another point has emerged to strengthen and foster user interaction by providing an interactive user interface that allows a user to modify the potential results provided by an automatic system in the case of incremental learning.

5. ***Knowledge modeling***:
   Recently, a set of open issues related to knowledge modeling have been tackled by the historical DIA community. The different open issues related to the knowledge generation and representation or modeling can help to:

   - Recreate the past by getting an overview of historical context,
   - Understand the interactions among various actors (e.g. centuries or epochs, writers, cultures),
   - Characterize the chronology and geography in historical events unfolding (e.g. constitution of new social classes).

   Various historical, social and political studies have evolved to characterize the dynamics of population distribution and the expansion of a population in the context of the 5CofM project, to determine the "social networks" linking historical collections and exploring the relationships among manuscripts and to identify the interactions over the centuries and among the writers and cultures in the context of IOW project, *etc.* Recently, researchers have addressed a challenge to build a database and ontology of facts extracted from HDIs for an intelligent information extraction, relevant summarization or knowledge discovery [86, 87].

It is worth noting that the research community is continuing to investigate and provide efficient functionalities suited to the users' needs such as easy image browser, relevant content-based retrieval, automatic computer-based access and analysis of DHBs, efficient computer-aided book or book page categorization tool, *etc.* For instance, Liang *et al.* [88] presented a generic framework which is called EMMEL, for historical manuscript image and data processing, visualization and analysis. The proposed framework provides a flexible description of the content of a historical manuscript and its meta-data on the one hand, media information enrichment (e.g. video, flash component) to the manuscript or a specific region of the manuscript. Grana *et al.* [89] proposed a complete system for analyzing automatically old documents and creating hyper linking between different epochs. The proposed system ensures a content enrichment of historical manuscripts with renovated contents. The augmentation or enrichment of cultural heritage documents with multimedia details is still challenging due to the high demand for automatic annotation and analysis of the digitized DIs with very little user intervention. By extracting images and keywords contained in their captions from DHBs, similar images from the Web were retrieved. Subsequently, the DHB

contents were enriched with new related contents. This enrichment ensures for example the comparison of how a historical site or an ancient monument described or illustrated in a DHB hundreds of years ago with how it is nowadays. In addition, it helps the user to find contemporary content connected (*i.e.* pictorial material from the Web) to the extracted components from the analyzed DHBs.

Nevertheless, these functionalities require a fine description and a high interpretation of book content. Thus, many approaches have been proposed to develop automatic tools for computer-assisted extraction of meta-data or signatures. These meta-data represent the content and/or structure of HDIs or part of them. The extraction of meta-data or signature definition are important issues for digital library development because they contribute to develop the different sub-fields and tasks related to the issues surrounding historical DIA such as word or graphic spotting, *etc.* In this context, LeBourgeois *et al.* [9] considered the meta-data extraction as a crucial task, since they suggested to design "intelligent" digitizers which can limit manual intervention and perform easy and high quality digitization of HDIs. The quality digitization of HDIs is adapted to the specificities of the digitized HDI content and structure, the user requirements (e.g. specified kind of information to extract and characterize), subsequent use of the digitized HDI or extracted HDI component, *etc.* Thus, in the context of our research project, DIGIDOC, we aim to design a computer-aided categorization tool, able to index or group digitized book pages according to several criteria, mainly the layout structure, graphical properties or typographical characteristics of the HDI content (*cf.* Section 1.1 in Chapter 1).

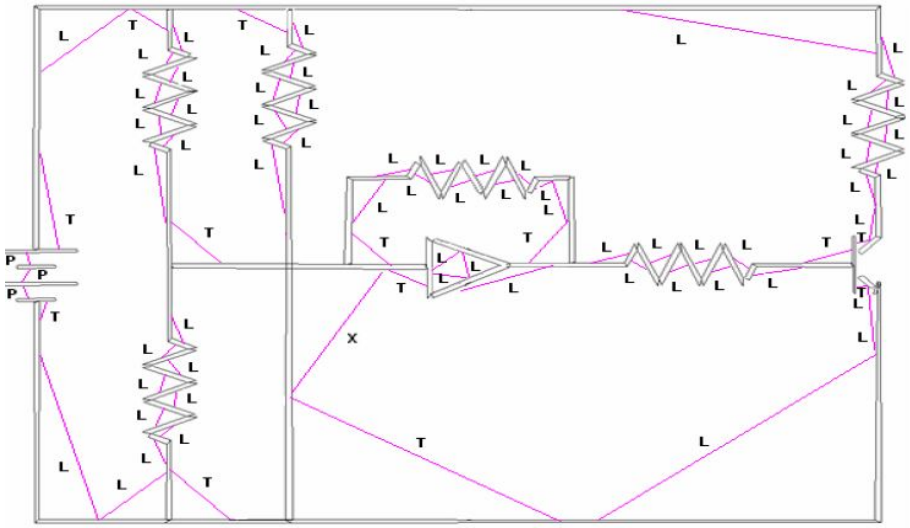


Figure 2.30.: Symbol spotting using a graph representation of graphical documents [81].

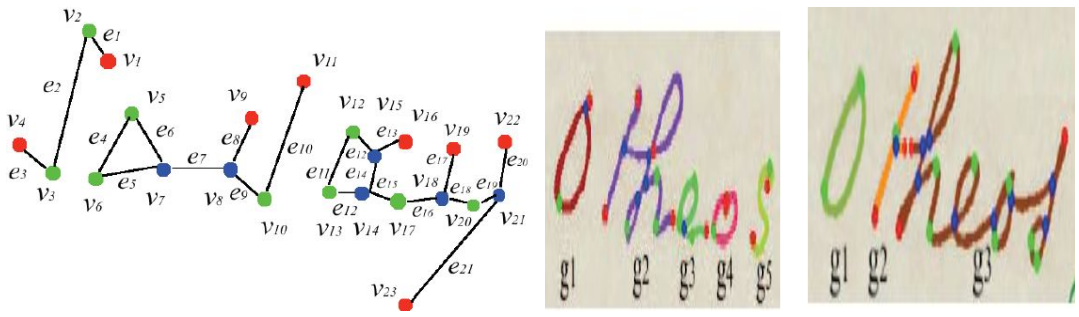


Figure 2.31.: A coarse-to-fine word spotting approach for historical handwritten documents based on the graph embedding and graph edit distance [82].

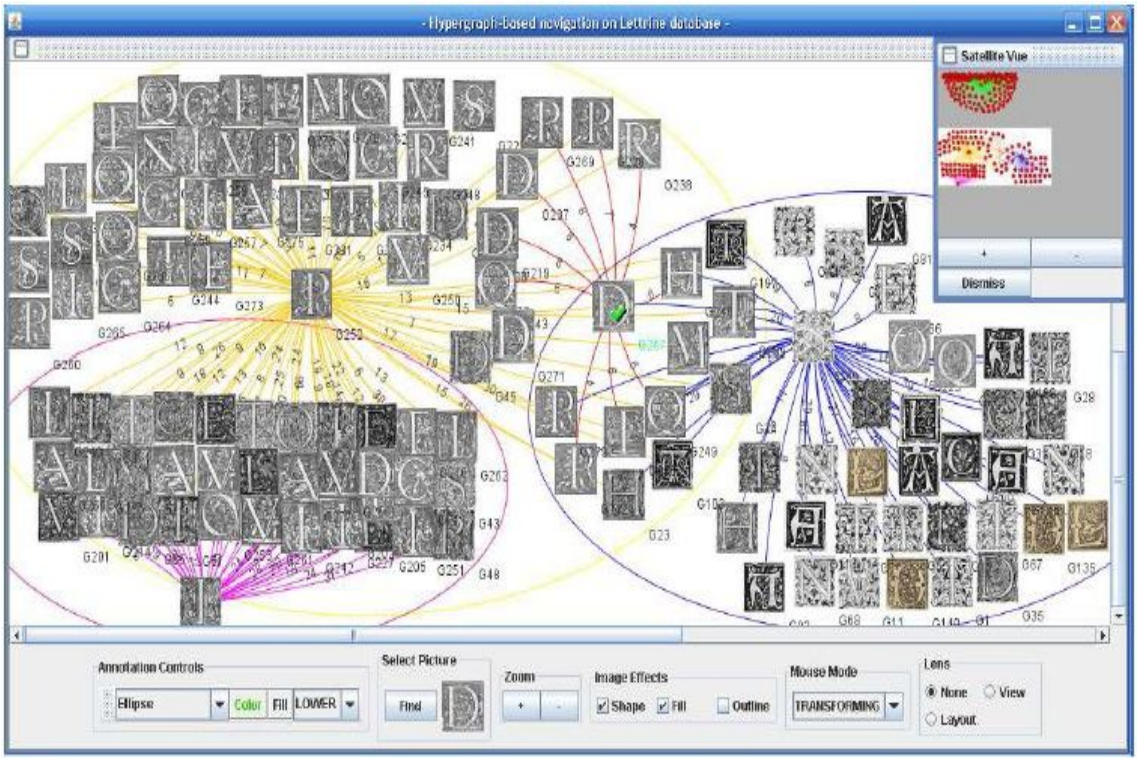


Figure 2.32.: Hyper-graph-based navigation on a drop cap database [15, 83].

Table 2.2.: A summary of the research projects dedicated to historical DIA.

Project	Goal	Task	Dataset	Tool	Result
A- Handwritten historical DIA and characterization					
Culture, inheritance and creation[32]	Handwriting classification [30]	-Identify the authors of a digitized corpus of handwritten HDIs from the 18^{th} and 19^{th} centuries, -Characterize and group together manuscripts written by the same author.	Montesquieu's and Flaubert's manuscripts[41]	Texture features	Ancient handwritten manuscripts of some famous French authors were evaluated since the particularities of handwritten HDIs have been covered (*i.e.* they contain multi-writer annotations or corrections and characterized by background noise and degradation such as background spots, delocalized folds, *etc.*).
GRAPHEM[30]	Automatic analysis of medieval writings [62, 63]	-Support palaeography experts in analyzing manuscripts, -Investigate and analyze the evolution of writing forms, -Develop efficient and automatic methods enabling accessing to manuscript contents based on word image similarity (*i.e.* word spotting and word retrieval).	Montesquieu's and Flaubert's manuscripts[41]	-Feature extraction, -Codebook, -Genetic algorithm, -Graph coloring, *etc.*	Numerous methods were proposed for handwritten content analysis, handwriting grapheme decomposition, grapheme analysis and classification *etc.* The developed algorithms can significantly help to transcript historical manuscripts and identify writing styles or writers.
Bovary[28]	Historical manuscript enrichment [47, 48]	-Characterize Flaubert's layout style, -Provide an on-line, structured access and browsing capabilities to an hyper-textual edition of "Madame Bovary" draft sets.	Flaubert's manuscripts[41]	-Bi-scale feature vector based on the pixel density measurement, -Hidden Markov models (HMM), *etc.*	Several solutions were proposed to ensure the segmentation of different page parts (e.g. words or parts of words) and the extraction of text lines or other objects of higher level (e.g. text blocks).

Table 2.2 – continued from previous page

Project	Goal	Task	Dataset	Tool	Result
Word spotting: indexing handwritten manuscripts[31]	Handwritten text and line retrieval [52, 40]	Word spotting for searching and indexing historical handwritten collections.	George Washington[34]	-Dynamic time warping (DTW), -Feature extraction, -K-means and agglomerative clustering technique, *etc.*	A Web-based retrieval system was developed for handwritten text.
5CofM[12]	Generation of a digital library of five centuries of marriages contained in marriage license books [45, 46, 67, 66]	-Word spotting, -Handwriting recognition.	-ESPOSALLES database[39] -BH2M database[40]	-Graph, -Feature extraction, -Structural information, *etc.*	Numerous approaches for handwritten text line segmentation, handwriting recognition and sequential word spotting in historical handwritten documents were proposed.
B- Graphical part indexing in historical heritage					
MADONNE[23]	Development of a toolkit to index heritage documents and categorize book pages [90, 29, 84]	-Decompose the information contained in lettrines into several layers to segment lettrines, -Design a lettrine CBIR system based on lettrine signature.	A lettrine dataset which is collected from the CESR[57]	Texture features	Texture-based signatures were produced for lettrine indexing
NaviDoMass[24]	Development of robust pattern recognition and analysis techniques supporting the particularities of HDIs (e.g. large variability of page layout, noise, degradation) [15, 51]	-Access to a rich library of rare Renaissance books, -Description, classification and indexing of HDI collections by their content.	Gray-scale lettrine images from the 16^{th} and 17^{th} centuries[42]	-Texture features, -Graph-based signature, -Ontology, *etc.*	-A structural-based framework to handle lettrine images was developed, -Generic solutions were proposed for lettrine indexing, recognition and classification.

Table 2.2 – continued from previous page

Project	Goal	Task	Dataset	Tool	Result
BVH[21]	Creation of a humanistic virtual library [49]	-Provide a public Web-portal accessibility to DHBs regardless barriers of time and place, -Index DHBs to ensure powerful new technological capabilities that enable users to search among titles, authors, dates and other different queries relative to the digitized books to retrieve a particular book or book element (e.g. graphical or textual parts).	85 rare DHBs which were collected from the CESR[57]	CC analysis technique	-A Web-based retrieval system interface of different kinds of graphics was proposed, -AGORA software was developed for the extraction and labeling of the graphical regions.
C- HDI layout analysis					
DEBORA[20]	Development of networked libraries by improving accessibility to the 16^{th} century books of Italy, France and Portugal [9, 32]	-Indexing, transmission, editing and annotation of book pages, -Compression of book pages for fast querying, navigation and downloading of required components of the logical structure and the physical layout of book pages.	Books from the 16^{th} century	CC analysis technique	-A complete processing chain was proposed for retrospective conversion, analysis, indexing, retrieval and compression of digitized Renaissance books, -A CAT was proposed to assist experts in manual transcription of Renaissance books. It is able to transcribe all printed documents regardless the typography and the language or alphabet used

Table 2.2 – continued from previous page

Project	Goal	Task	Dataset	Tool	Result
BVH[21]	Management of book content description [72]	-Index DHBs by extracting and structuring meta-data, -Extract table of contents or graphics, -Transcript the textual blocks	85 rare DHBs (Vesalius's manuscripts[42]) which were collected from the CESR[57]	CC analysis technique	An interactive HDI layout analysis and segmentation tool which is called AGORA[63], was developed. AGORA is a user-driven annotation tool which performs HDI layout analysis.
DMOS[25]	Development of generic document recognition method [73, 59, 60]	-Make handwritten archives documents accessible to public with a generic system of DIA, -Extract automatically structure from quite damaged military forms of the 19^{th} century, found in French archives, -Provide an automatic access to military form pages by handwritten content recognition (*i.e.* retrieve the right documents according to a textual request on the last name) after locating precisely the handwritten last name cell.	$88,745$ old civil status registers and military forms of the 19^{th} century	-EPF, -Parser, *etc.*	-A generic recognition method of 2-D structures was proposed, -The FormuRead software was developed to extract automatically structure from quite damaged military forms of the 19^{th} century, found in French archives, -A platform for managing all annotations produced by DIA was proposed to make handwritten archives documents accessible to public.

Table 2.2 – continued from previous page

Project	Goal	Task	Dataset	Tool	Result
METAe[26]	Development of a set of tools able to digitize and analyze books and journals with a minimum of effort and a maximum of automation and effectiveness [74]	-Provide an automated and structured conversion of printed ancient documents of the 19^{th} and 20^{th} centuries into digital formats, -Recognize and characterize of the physical and logical document or book structure through the generation of image meta-data and character recognition using OCR, -recognize specific fields such as page numbers, titles, size of fonts, page footnotes, *etc.*	Printed ancient documents of the 19^{th} and 20^{th} centuries	Structural meta-data	A software program known as DocWorks was developed. It offers an automated and structured conversion of printed ancient documents of the 19^{th} and 20^{th} centuries into digital formats. For easy access and searchability, DocWorks ensures the automatic recognition and description of the physical and logical document or book structure through the generation of image meta-data and character recognition using OCR. It can recognize specific fields such as page numbers, titles, size of fonts, page footnotes, *etc.*
PlaIR[27]	Development of a platform for indexing and searching of multi-domain and multi-purpose information from a set of digital library resources [75]	Automatic or semi-automatic analysis of digital library resources	Archives of the "Journal of Rouen" newspapers from the years 1768 to 1848	Conditional random field (CRF) model	An on-line research and consultation application which is called PIVAJ, was developed to offer a world-wide access to the digitized archives of the Journal of Rouen

Table 2.2 – continued from previous page

Project	Goal	Task	Dataset	Tool	Result
-HisDoc[13] -HisDoc2.0[14]	Text localization, script discrimination and scribe identification in historical manuscripts [41, 42, 43, 34, 53, 56, 55, 54, 57, 4, 77, 78]	-Historical DIA: HDI enhancement by modeling, understanding and eliminating the noise and degradation and HDI layout analysis by describing and characterizing the layout and content of HDIs, -Handwritten text recognition: produce a fully automatic and robust segmentation and transcription system of text line images, -Information retrieval: implement a search engine for transcriptions.	IAM-HistDB[33]	-HMM, -Dynamic multi-layer perceptron (MLP), -Support vector machine (SVM), -Gaussian mixture models (GMM), -Color and texture feature extraction, -Pixel neighborhood, -Adapted greedy forward selection and genetic selection, *etc.*	A complete system for HDI layout analysis, handwritten text recognition and information retrieval was proposed.
5CofM[12]	Retrieval of textual information from huge data collections [64, 65, 66]	-Structured document segmentation, -Segmentation of touching lines in historical handwritten documents.	5CofM dataset[53]	-RLF features, -Graph, -Bi-dimensional extension of stochastic context-free grammars, *etc.*	-Good structured document segmentation results were obtained, -High performance for segmenting touching lines in historical handwritten documents was obtained comparing other state-of-the-art methods even the DIs contain skewed, multi-oriented, touching or overlapping lines.

Table 2.2 – continued from previous page

Project	Goal	Task	Dataset	Tool	Result
D- Historical collection modeling and representation					
MADONNE[23]	HDI indexing [79, 1]	-Identify the similarities concerning the collection structure by generating a relevant model summarizing each analyzed book, -Categorize the book pages.	A number of DHB pages (Vesalius's manuscripts[42]) collected from the CESR[57]	Texture features	A pixel-based classification approach was proposed as the first step in a work of indexing HDIs. It can help to characterize and identify the similarities concerning the collection structure (e.g. layout, typography) by generating a relevant model summarizing each analyzed book.
IOW[15]	Study and analyis of the history of human-environment interaction in the Indian ocean world [80]	-Process and understand ancient manuscripts, -Determine the "social networks" linking historical collections and exploring the relationships among manuscripts, -Characterize the interactions over the centuries and among the writers and cultures.	Small collection of original manuscripts collected from the IOWC archives[74]	-Multi-spectral imaging, -Spatial, spectral, sparse and graph-based representations of visual objects, -Directed graphical models, HMM, undirected random fields and spatial relations models, -VLP, *etc.*	A data-driven network-oriented analysis framework of historical manuscripts based on the VLP for pattern analysis was proposed. It combines the visual class (images and manuscripts) and conceptual class (phrases and manuscripts) to determine and characterize the "social networks" linking ancient manuscripts (from the low-level relations of patches, excerpts, *etc.* to the high-level relations of manuscripts, collections, writers, *etc.*).

Chapter 3.

From document image analysis to historical document image analysis

This chapter outlines related works on document image analysis, by firstly detailing the classical approaches. Then, the texture-based methods proposed in the literature are described, with a particular focus on those related to document image analysis and historical document image analysis.

Contents

3.1. Introduction

From the 1980 onwards, several surveys and comparative studies of the basic concepts and techniques concerning the two areas of document processing: DIA and DI understanding have been discussed [91, 92]. Tang *et al.* [91] stated that a DI has two structures: geometric/layout structure and logical structure. By extracting the geometric or layout structure from a DI, the DIA techniques are involved. When the geometric structure is mapped into the logical one, the DI understanding approaches are examined. In this work, we are only interested in the DIA issues.

DIA has been a thriving topic of major interest of many researchers and one of the most explored fields in image analysis. Nagy [71] summarized the different studies and contributions achieved on different sub-fields and tasks of DIA (e.g. pre-processing, character recognition, page decomposition, graphics recognition). DIA consists in dividing a DI layout according to the nature of the extracted structure such as separate text from non-text regions or partition text into columns, text blocks, lines, words, *etc.* It starts by segmenting a DI in order to find and classify homogeneous regions or zones, such as graphic and textual regions [6]. Finding graphic regions can be used to segment and analyze the graphical part in historical heritage such as the drop caps [29], while determining text zones can be used as a pre-processing stage for character recognition [93], text line extraction [94], handwriting recognition [54], *etc.*

Beyond this point, this chapter outlines the related works on DIA. The remainder of this chapter is organized as follows: Section 3.2 presents a brief description of the main basic concepts and techniques, challenges, issues related to DIA. Section 3.3 reviews related works on the segmentation, characterization and analysis of the block contents of the DI. The classical and texture-based methods proposed in the literature are described, with a particular focus on those related to DIA and historical DIA. Our discussion and conclusions are presented in Section 3.4.

3.2. Definitions and challenges

Several DIA studies have been conducted and reported in the literature in order to characterize the DI layout with the result of structuring it into three different levels [92, 95]:

- ***Physical level***:
  The physical level specifies both the typography and document organization. The typography arranges the information style (e.g. fonts, colors, lines, frames) and the form layout (e.g. line spacing and alignment). The document organization sets the layout of all the visual elements (e.g. characters, words, lines, blocks, columns, non-text regions) contained in a DI and the topological or spatial relationships between those elements (e.g. hierarchy, inclusion, neighborhood position).

- ***Intermediate functional level***:
  The intermediate functional level represents a physical interpreted one which ensures the recognition of the logical structure. This logical structure is obtained by recognizing the role of each component of the physical structure based on *a priori* knowledge on the typography, layout, *etc.*

- ***Logical level***:
  The logical level aims to interpret and recognize the different parts that compose a DI and specify the logical relationship between them. The recognition of logical structure is necessarily guided by a model from either the physical structure or functional structure.

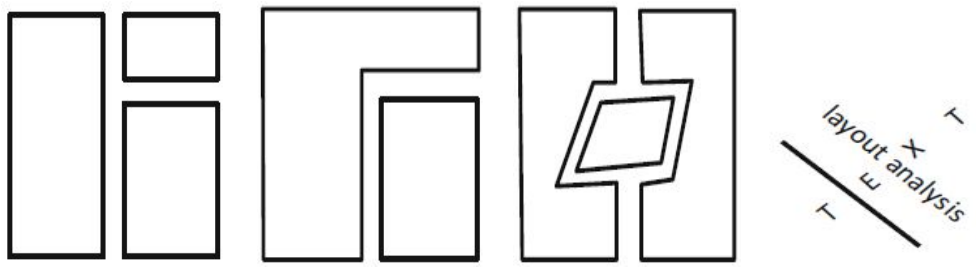
In this work, we focus only on the first level, *i.e.* the physical level. In the literature, different DIA approaches have been presented for segmenting, characterizing and analyzing the extracted block contents from the DI layouts. Kise [5] categorized the DI layouts into four classes (*cf.* Figure 3.1):

- ***Rectangular layout***:
  A DI has a rectangular layout, when all its components are circumscribed by non-overlapping rectangles and when the sides of its components are parallel or perpendicular to the DI borders (*cf.* Figure 3.1(a)).

- ***Manhattan layout***:
  A DI has a Manhattan layout, when its components can be circumscribed by overlapping (*i.e.* having concave shapes) and non-overlapping rectangles and when the sides of its components are parallel or perpendicular to the DI borders (*cf.* Figure 3.1(b)). The rectangular layout is a particular case of the Manhattan one. Newspapers are classic examples of the Manhattan layout.

- ***Non-Manhattan layout***:
  A DI has a non-Manhattan layout, when its components can be circumscribed by non-overlapping regions and when the sides of its components are neither parallel nor perpendicular (*i.e.* slant sides) to the DI borders (*cf.* Figure 3.1(c)). Magazines with larger figures and pictures are typical examples of the non-Manhattan layout.

- ***Overlapping layout***:
  A DI has an overlapping layout, when DI blocks or components intersect each other (*i.e.* superimposition of information layers), when pixels of one DI component are adjacent to those of others or when there is no clear distinction between the foreground and background pixels (*cf.* Figure 3.1(d)). Modern publications such as advertisement or historical documents are standard examples of the overlapping layout.



(a) Rectangular (b) Manhattan (c) Non-Manhattan (d) Overlapping

Figure 3.1.: Four classes of DI layouts: rectangular, Manhattan, non-Manhattan and overlapping [5].

Several scientific works in contemporary DIA have described several relevant approaches enabling multiple forms of indexing based on content analysis of DIs. For instance, the text in scanned DIs is automatically converted to editable text by OCR and stored afterwards in digital databases. This would both ensure access to the meaning of words in the pages, and easy and quick search for occurrences of words in the text. In this context one can also mention the retrospective conversion tools allowing the access to DI layout on the one hand, and indexing illustrations or pictures contained in DIs on the other hand. Thus, current systems for categorizing digitized DIs are based on several criteria, such as the textual content by applying OCR or by using the interest point detection approach. For instance, Augereau *et al.* [96] classified industrial DIs by combining visual and textual features. The visual features were extracted with the bag of words (BoW) technique, while the visual ones were extracted with the bag of visual words (BoVW) approach. Bouguelia *et al.* [97] presented a learning approach for a DI classification task in an industrial context. They used a BoW representation of DI to classify administrative DIs by their topics. The analyzed DI was firstly processed by an OCR. Then, it was represented as a BoW which is

a sparse feature-vector containing the occurrence counts of words in the analyzed DI. Klein *et al.* [98] presented smartFIX as a document analysis and understanding system developed during the "Adaptive READ"[1] research project. The proposed system integrates the DCAdmin module (*i.e.* a graphical interface for the training of a classification component which is called mindaccess) which helps to supervise the learning of semantics of DI content in order to classify the medical bills and prescriptions with significantly similar content. At the industry sector level, several information technology (IT) service companies, particularly those involved in DIA fields (e.g. ITESOFT[2], A2IA[3]), offer many services and toolkits dedicated to DI classification. For example, the A2IA company developed the "A2IA Document Reader"[4] software to classify all types of paper document and incoming mail, regardless of their structure or content. The developed tool proceeds with the extraction of all occurrences of searched keywords. It is adapted to all kinds of script (e.g. machine-print, hand-print, cursive). Moreover, Google sponsored an OCR engine which is called Tesseract, able to extract and recognize automatically text through layout analysis and image recognition modules.

Nevertheless, the transposition of these tools for historical DIA, that are dedicated initially for contemporary DIA, is not straightforward. Grana *et al.* [89] stated that, despite the OCR-based methods have yielded reliable results for contemporary DIA, analyzing the HDIs by separating textual regions from the graphical ones is still more challenging. Indeed, these tools for performing the historical DIA tasks have poor performance due to many particularities of HDIs. Kise [5] stated that DIA of pages with constrained layouts (e.g. rectangular, Manhattan) and clean DIs has almost been solved while historical DIA is still an open problem due to their particularities.

HDIs have the following particularities:

- ***Properties***:
  - Large variability of page layout,
  - Complicated and complex page layout (e.g. several columns with irregular sizes, dense printing, irregular spacing, marginal notes),
  - Random alignment,
  - Use of specific and multiple fonts and illustration styles,
  - Large variability of editorial style and logical structure,
  - Presence of embellishments,
  - Irregular spacings (e.g. between characters, words, lines, paragraphs or margins),
  - Overlapping object boundaries,
  - Varying text column widths,
  - Interspersed graphics,
  - Frequent use of different kinds of graphics (e.g. ornaments, drop caps, frames, embellishments, portraits),
  - Graphical illustrations and their legends in different and variable locations,
  - Text in different orientations,
  - Presence of location indicators (e.g. line numbers, page numbers, catchwords),
- ***Life cycle***:

[1] http://www.dfki.de/pas/kmc/adread-e.html
[2] http://www.itesoft.co.uk/
[3] http://www.a2ia.com/en
[4] http://www.a2ia.com/en/document-classification

  - Noise and degradation caused by copying, scanning or aging (e.g. yellow pages, ink stains, mold or moisture, faded out ink, uneven lighting due to folded, corrugated parchment or papyrus),
  - Superimposition of information layers (e.g. stamps, handwritten notes at the margins, noise, back-to-front interference, ink that was bleeding through, historical spelling variants),

- ***Digitization***:
  - Page skew,
  - Scanning defects (e.g. curvature, light),
  - Presence of black borders, *etc.*

Figure 3.2 illustrates some particularities of HDIs, such as the superimposition of information layers (e.g. stamps, handwritten notes, noise, back-to-front interference, page skew) which were collected from the French digital library Gallica[3] [99, 8].

Moreover, the OCR-based and contemporary DIA methods are hindered by many issues related to the OCR and retrospective conversion performance. In addition, they require burdensome and complex processing due to the mentioned particularities of HDIs. Indeed, this way of using several criteria for the textual content to categorize DHB pages is beyond the scope of this work. This points to the need for further reflection on finding an automatic and parameter-free feature-based tool, able to index or group DHB pages according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the HDI content. This tool should be independent of the layout and content of the analyzed DHB pages (e.g. HDI layout, typeface, font size, orientation, digitizing resolution) and applicable to a large variety of DHBs. Moreover, we need to refine the focus on extracting and analyzing optimal features (other than textual-based features), to provide a rich and holistic description of the layout and content of the analyzed DHB pages.

Thus, processing this kind of document is not a straightforward task and usually includes several stages: pre-processing, analysis, characterization and recognition [100]. For the problem of historical DIA, the main challenge is to analyze HDIs and to characterize their layouts and contents under significant degradation levels and different noise types and with no *a priori* knowledge about the layout, content, typography, font styles, scanning resolution or DI size, *etc.*

Kise [5] stated that several criteria can be used to categorize the state-of-the-art methods for page segmentation in DIA. He defined three criteria which are:

- ***Object to be analyzed*** (*i.e.* foreground or background):
  When documents are printed in black and white, it is trivial to distinguish between the foreground and background. Indeed, it is usually considered that the black parts in documents printed in black and white correspond to the document foreground (e.g. characters, figures), while the white parts represent the document background. Nevertheless, it is not so simple to separate the foreground from the background when documents are not printed in black and white (*i.e.* both the foreground and the background can have similar colors when documents are printed in color) or for DIs with overlapping layout (*i.e.* background of a page component can be considered as the foreground of another page component). Analyzing either the foreground or background of a DI consists in analyzing the different components that belong to either the foreground or background. Kise [5] confirmed that foreground/background separation is a non-trivial task particularity for gray-level and color DIs and for DIs with overlapping layout. Thus, he pointed out that the choice of the appropriate analysis primitives is crucial.

- ***Primitive of analysis*** (*i.e.* pixels, superpixels, CCs, *etc.*):
  Analyzing either the foreground or background of a DI is processed by investigating their respective primitives. A primitive is an element of a DI that belongs to either the foreground

Figure 3.2.: Illustration of some particularities of HDIs, such as the superimposition of information layers (e.g. stamps, handwritten notes, noise, back-to-front interference, page skew) collected from the French digital library Gallica[3].

or background. Pixels, superpixels and CCs are the most used primitives in both foreground and background analysis.

- ***Pixel*** is the smallest manageable element of a DI.
- ***Superpixel*** becomes a consistent alternative of using a rigid structure of pixel grid. Indeed, a superpixel is a set of pixels sharing similar characteristics (e.g. texture cues, contour, color) into a significant polygon-shaped region.
- ***CC*** is an important primitive when documents are binary images and DIs with non-overlapping layout. A CC is defined according to the basic concept of graph theory which is the connectivity. The connectivity illustrates the interconnections of pixels in images to their neighbors. In general, 8-connectivity is used for black pixels, while 4-connectivity is often employed for while pixels to avoid the crossing connections.

For analyzing DIs with overlapping layout, the most fundamental and used primitive is pixel,

while for gray-level and color DIs, the superpixel approach has been recently investigated and examined. The CC analysis technique can be used in the case of gray-level and color DIs, when the results of the conversion of gray-level and color DIs to binary ones still reasonable.

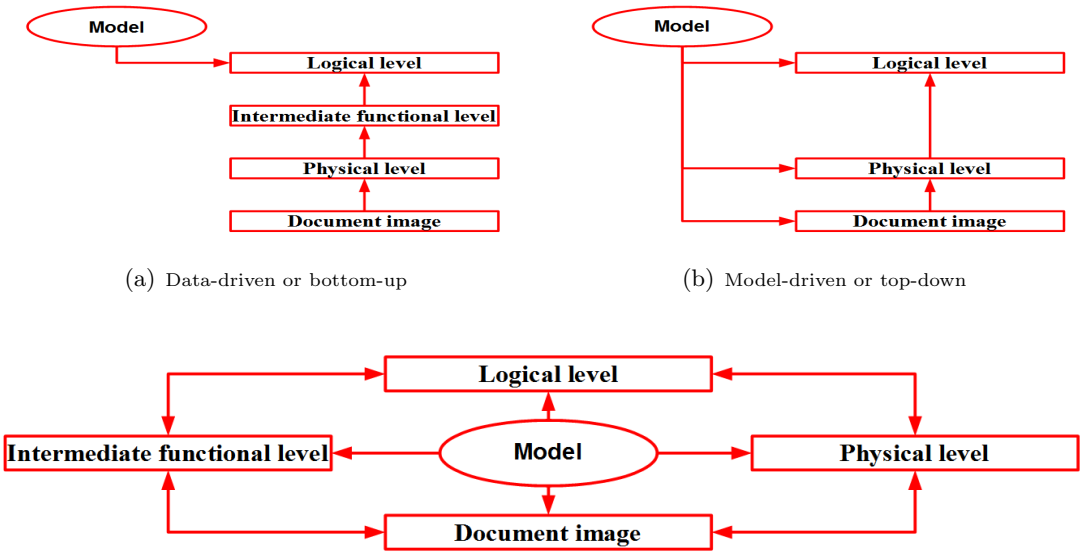
- ***Strategy of analysis*** (*i.e.* bottom-up, top-down and hybrid):
  There are three categories of analysis strategies [92, 95]:

  - ***Data-driven or bottom-up strategies of analysis***:
    They do not include (or little) knowledge of a document model (*cf.* Figure 3.3(a)). Those strategies of analysis are based on low-level data mining of pixels (e.g. color, position). For example, the run-length smearing algorithm (RLSA), also known as the constrained run-length algorithm (CRLA), studies the spaces between black pixels in order to link neighboring black areas [101, 102]. O'Gorman [103] proposes a method for page layout analysis based on the extracted CCs and the nearest neighbor (NN) clustering of page components. There are certain limitations of this category of analysis strategy: firstly, the proposed methods in this category of analysis strategies are based on the definition of complex criteria and rules. Secondly, they are sensitive to noise and not robust to slanted texts. Thus, they are suitable for DIs whose areas are clearly demarcated and rectangular. Furthermore, the pertinence of this category of analysis strategy depends on the particular layout and idiosyncrasies of DIs.

  - ***Model-driven or top-down strategies of analysis***:
    They are guided by a document model (*cf.* Figure 3.3(b)). Often used for well-defined and invariant structured DIs, those strategies of analysis are based on strong *a priori* knowledge to guide the segmentation and recognition. For instance, the recursive XY-CUT (RXYC) algorithm consists in computing the horizontal and vertical projection profiles (*i.e.* corresponding to the sum of the pixels along the horizontal and vertical axes) and iteratively splitting them into smaller ranges until a condition about hollow projections or valleys in the projection profile histograms (corresponding to inter-line white spaces) has been satisfied. This requires the definition of criteria for cutting (and possibly fusion). In addition, it assumes that the input DIs are not skewed [104]. Although the model-driven analysis strategies are generally faster, they are not well-adapted to complex layout and skewed DIs.

  - ***Hybrid or mixed strategies of analysis***:
    They combine data-driven and model-driven strategies of analysis (*cf.* Figure 3.3(c)). A split-and-merge strategy is an example of a strategy of analysis [105]. Kida *et al.* [106] proposed to segment an image by a sequence of horizontal and vertical projections, then they used connectivity analysis for document recognition system for office automation. Chen *et al.* [107] proposed a method based on whitespace rectangle extraction and grouping to form text lines and afterwards text blocks. The proposed method proceeded by analyzing the extracted foreground CCs and filtering the gaps between the horizontally adjacent CCs. Lazzara *et al.* [108] discriminate text from non-text regions by analyzing the extracted foreground CCs and using the object alignment and morphological algorithms. By combining tools from data-driven and model-driven strategies of analysis, hybrid approaches combine the high speed of the model-driven methods with the robustness of the data-driven ones. They can deal with a wide variety of DIs and cope with complicated page segmentation problems, but many parameters and thresholds must be adjusted.

The standard flowchart of an analysis strategy is often represented in the form of a sequential process that goes from one level to another (*cf.* Figure 3.3) [95].



(a) Data-driven or bottom-up

(b) Model-driven or top-down

(c) Hybrid

Figure 3.3.: Three categories of analysis strategies: the data-driven or bottom-up, model-driven or top-down and hybrid.

3.3. Related works

In the literature, different algorithms have been presented for the segmentation, characterization and analysis of the block contents of the DI physical layout. Okun and Pietikäinen [6] classified the methods developed for DIA into two categories: texture and non-texture-based. They analyzed the existing texture-based methods for document analysis and briefly compared them to the non-texture-based ones. Texture-based approaches consider a document as a set of different textured classes while the non-texture-based ones involve different image processing techniques, such as CC analysis [103], split-and-merge algorithm [105], *etc.*

Kise [5] categorized the existing DIA algorithms into two classes: the methods dealing with non-overlapping layout and those dealing with overlapping layout. Then, the DIA algorithms dealing with non-overlapping layout can also be divided into two categories: foreground analysis and background analysis methods. The foreground analysis methods can also be classified into three sub-categories: projection-based, smearing-based and CC-based methods. Among the background analysis algorithms, we cite, for example, the shape-directed covers [109, 110], white tiles [111], Voronoi diagram [112, 113], white space analysis [114], *etc.* In this work, we are interested in foreground analysis methods and particularly in DIA for page segmentation. Those methods may be divided into two types: the classical approaches which are based on image analysis tools and strong *a priori* knowledge and those based on texture analysis descriptors [18, 95].

3.3.1. Classical approaches

The classical DIA approaches are devoted to contemporary DIs and are significantly widespread in the literature because those methods are based on a strong *a priori* knowledge such as the repetitiveness of document structure in a corpus. This family of document structure analysis and page segmentation methods combine image analysis tools and strong *a priori* knowledge.

3.3.1.1. Categories of classical approaches

The usual way of presenting these approaches is to classify them into three categories [5]: the projection-based, smearing-based and CC-based methods.

1. ***Projection-based methods***
   The projection-based methods analyze the projection profiles of a DI. Figure 3.4 illustrates an example of the horizontal projection profile of a text black in a DI. By analyzing the horizontal (resp. vertical) projection profiles, the length of horizontal (resp. vertical) gaps below the horizontal threshold τ_h (vertical threshold τ_v) can identify the horizontal (resp. vertical) borders of different page components and subsequently ensure DIA. Indeed, blocks of a DI can be split vertically (resp. horizontally) by determining the vertical (resp. horizontal) cuts which correspond to wider gaps of horizontal (resp. vertical) projection profiles.

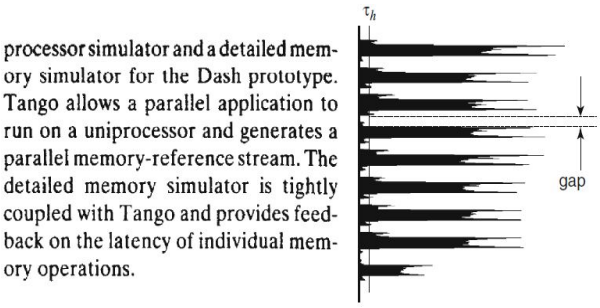
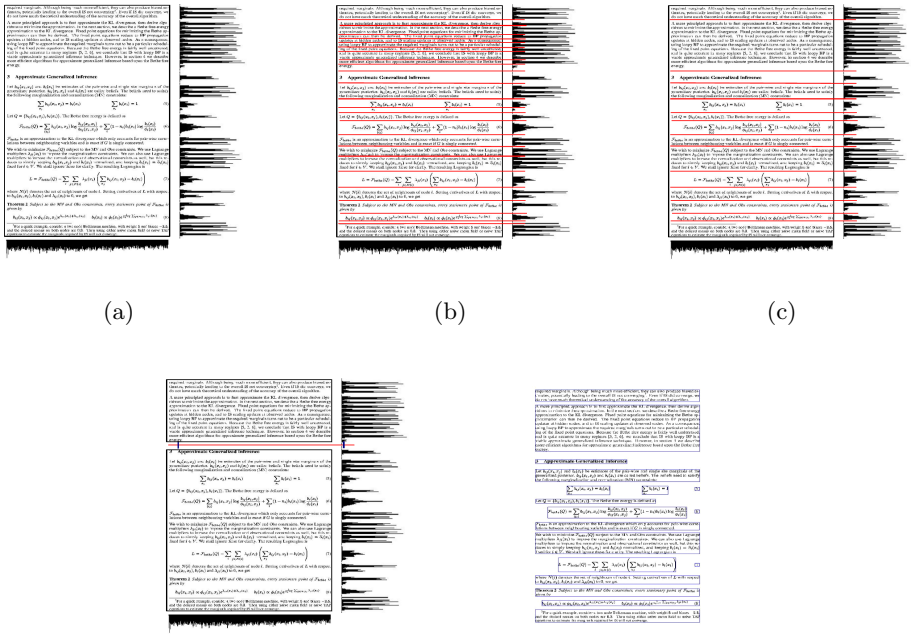


Figure 3.4.: Illustration of the horizontal projection profile of a text black in a DI [5].

- ***RXYC*** is a top-down DI segmentation technique that decomposes a DI recursively into a set of rectangular blocks [104]. It detects and separates all rectangular blocks separated by white spaces. The idea is to recursively apply the same algorithm to an area to obtain two regions, while a condition about hollow projections is not satisfied (selecting the widest cut first, until no cut wider than a certain minimum threshold is reached. It consists in analyzing the horizontal and vertical projection profiles of the whole DI by summing up all the pixels in a line to the sides of the DI. Then, by generating the white space density graph from the produced projection profiles and extracting the peaks from them, the cuts of the DI can be defined and DI can be segmented into rectangular blocks. The RXYC algorithm is well-suited to printed DIs (e.g. newspapers) where the document is well-structured and divided into rectangular blocks. However, this algorithm is not well-adapted to varied, complicated and complex DI layout or if the DIs are skewed. An example of the application of the RXYC algorithm is illustrated in Figure 3.5. The final segmentation result of the application of the RXYC algorithm on a contemporary DI shows few errors (e.g. regions are split, and lines are merged with footnotes) (*cf.* Figure 3.5(e)) [5].
- ***Syntactic segmentation*** is a top-down method based on X-Y tree representation for extracting alternating horizontal and vertical projection profiles from nested sub-blocks of scanned page images for syntactic segmentation and labeling of digitized pages from technical journals (*cf.* Figure 3.6) [115]. The proposed method has the advantage of backtracking to correct mistakes. It is only applied on families of technical DIs that share the same layout conventions (e.g. IBM Journal of Research, Development and IEEE Transactions on PAMI).
- ***White streams*** is a top-down method proposed by Pavlidis and Zhou [105] for page segmentation and classification (*cf.* Figure 3.7). It is based on the detection and analysis

[5] `www.cs.utoronto.ca/~klaven/ttpres/ltc-pres.ppt`



(a) (b) (c)

(d) (e)

Figure 3.5.: Illustrative example of the application of the RXYC method for page segmentation. Figure (a) illustrates the horizontal and vertical projection profiles. Figure (b) depicts the identification of the hollow projections or valleys in the projection profile histograms (corresponding to inter-line white spaces) produced by the horizontal projection profile. Figure (c) shows the selection of the hollow projections by eliminating those narrower than some threshold. Figure (d) presents the selection of the widest cut and the application of the same algorithm to the two new obtained regions. Figure (e) shows the final segmentation result[5].

of large areas of white spaces. These white spaces form a continuous stream by extracting, analyzing and grouping the physical properties (the white margins, white inter-line, white inter-character space, *etc.*). The proposed algorithm is particularly well-suited to DIs that contain rectangular and clearly demarcated blocks (e.g. newspapers, technical documents).

- ***Hough transform*** is a bottom-up technique that is applied on the extracted CCs for text string separation from mixed text/graphic images [116] and textual image analysis [117]. The Hough transform is considered as a generalization of the projection profile method of detecting document skew. The Hough technique is applied for text string separation from mixed text/graphic images (*cf.* Figure 3.8) [116]. The proposed algorithm is based on the CC extraction and Hough transform application to group together the extracted CCs into logical character strings by using simple heuristics based on the text string characteristics. It is relatively independent of text font styles, sizes and orientations. Moreover, it adapts to changes in text characteristics within the DI. Nevertheless, its performance depends on the conformity of the provided constraints and defined heuristics on the analyzed DI characteristics.

2. ***Smearing-based methods***
   Unlike the projection-based methods which investigate the white spaces (*i.e.* gaps) between the DI components, the smearing-based ones extract, examine and fill the space within each DI components. The morphology-based and RLSA methods are two representative techniques

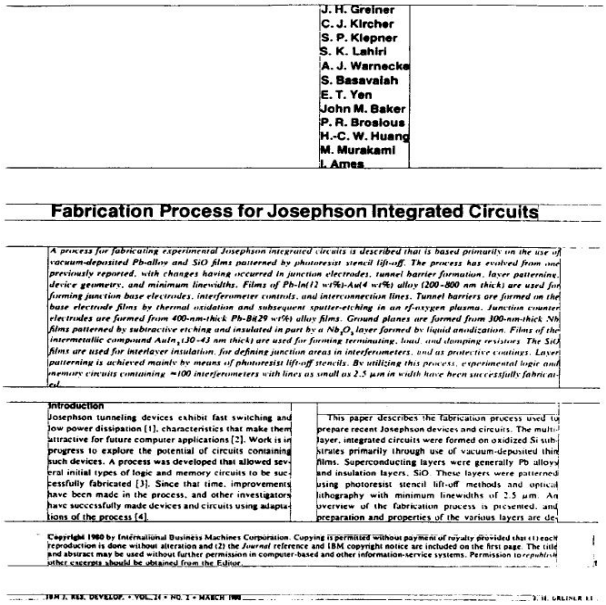


Figure 3.6.: Illustration of the application of the method based on X-Y tree representation for syntactic segmentation of a title page from the IBM Journal of Research and Development [115].

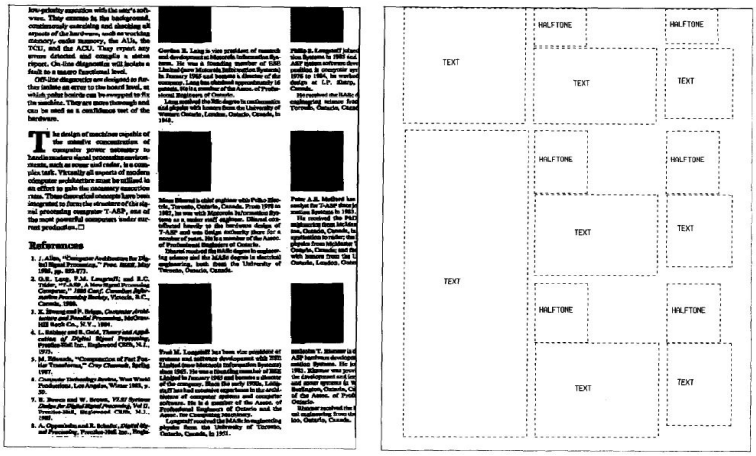
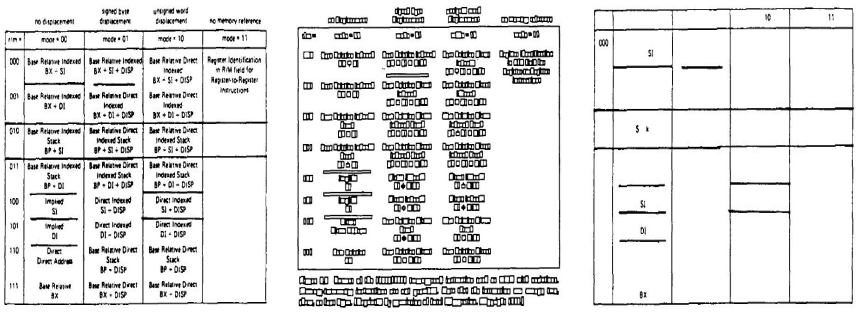


Figure 3.7.: Illustration of the application of the algorithm of white streams for page segmentation and classification [105]. The dotted lines mark the boundaries of the final column blocks. The labels are obtained by the classification process in the second stage.

of the smearing-based methods.

- ***Morphology-based methods*** are pixel-based bottom-up methods which are mainly used to identify text lines when their directions and the layout structure are known in advance. A series of morphological operations is applied to obtain an efficient and reliable segmentation [118, 119]. The morphology-based methods are based on using two fundamental operations of mathematical morphology which are the dilation and erosion.
  - ***Dilation***: $I \oplus E = \bigcup_{a \in E} I_a$ sets out to dilate the foreground of I by the defined amount in E.
  - ***Erosion***: $I \ominus E = \bigcap_{a \in E} I_{-a}$ sets out to erode the foreground of I by the defined amount in E.



(a) (b) (c)

Figure 3.8.: Illustrative example of the application of the Hough technique for text string separation from mixed text/graphic images. Figure (a) illustrates the original image. Figure (b) depicts the extraction of the CCs. Figure (c) shows the output result after a text string separation task [116].

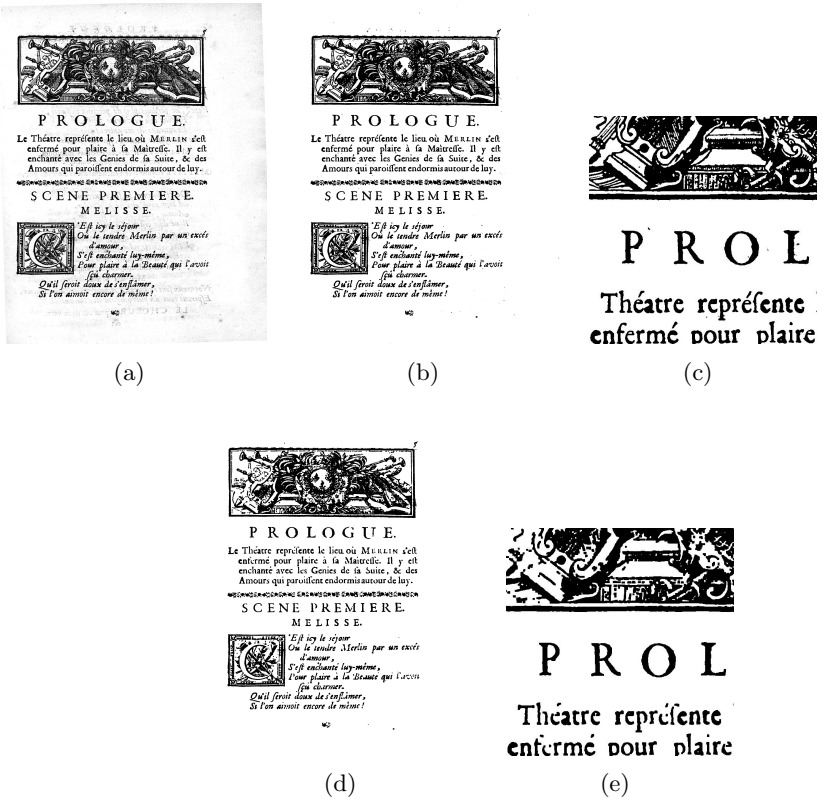
where I is an input image, E is a structuring element and I_a is the translation of I along the pixel vector a. $\oplus$ and $\ominus$ denote the dilation and erosion operators.

Based on these two fundamental operations of mathematical morphology, two other operations are deduced which are the opening and closing.

- ***Opening***: $I \circ E = (I \ominus E) \oplus I$ is aimed at removing small objects respecting E.
- ***Closing***: $I \bullet E = (I \oplus E) \ominus I$ is aimed at filling small gaps respecting E.

where $\circ$ and $\bullet$ denote the opening and closing operators. An illustrative example of the application of the closing operation with a 5×5 rectangular structuring element on a HDI is shown in Figure 3.9. We can see that characters are erased while gaps are filled in graphic blocks, and few noise pixels are removed (*cf.* Figure 3.9(e)). The obtained result (*cf.* Figure 3.9(d)) demonstrate that by using an isotropic structuring element, text lines and blocks can not be grouped and afterwards it is not a straightforward task to identify text lines and blocks effectively. However, the skewed DIs can be analyzed with the morphology-based methods when the structuring element is isotropic.

Bloomberg [118] segmented DIs into text and halftone components using multi-resolution morphology. He suggested to adapt the use of the fundamental operation of mathematical morphology according to the content type. Since the distance between pixels in halftones is smaller than that between characters, the closing operation is applied on halftones to fill the gaps in their pixels and obtain large blobs representing part of halftones, while the opening is applied after on text blocks to erase characters. Thus, by using relatively large structuring elements, large blobs are obtained and are considered as seeds helping to guide the identification of pictures (*cf.* Figure 3.10(c)). To overcome the limitation of the computational burden of a morphology-based method (caused by using large structuring elements), Bloomberg [118] used the multi-resolution morphology which is called the threshold reduction by applying it many times and varying the threshold value on each step (*cf.* Figure 3.10(b)). The threshold reduction consists in converting 2×2 pixels into one pixel using a threshold deducing from the sum of 2×2 pixel values. Bukhari *et al.* [119] proposed improvements to the text/halftone segmentation algorithm described by Bloomberg [118] by making it a general text and non-text image segmentation approach where non-text components (e.g. halftones, drawings, maps, graphs) (*cf.* Figure 3.10). They proposed to introduce a task based on hole-filling step in the Bloomberg's algorithm [118] and they showed accurate non-text mask for drawing type components (*cf.* Figure 3.10(h)).



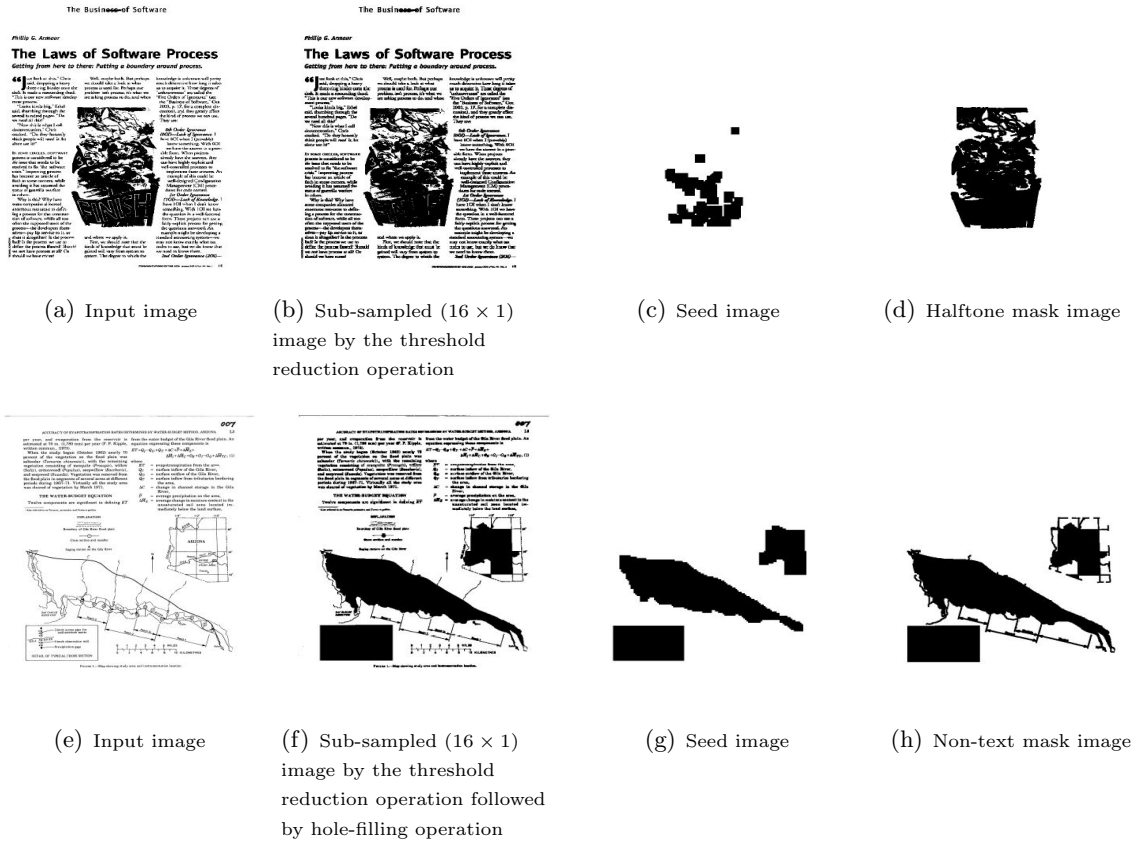
(a) (b) (c)

(d) (e)

Figure 3.9.: Illustration of the application of the closing operation with a 5×5 rectangular structuring element on a HDI. Figure (a) illustrates the original HDI. Figure (b) illustrates the binarized HDI. Figure (c) shows a zoomed region of Figure (b). Figure (d) depicts the result of the application of the closing operation with a 5×5 rectangular structuring element on a HDI. Figure (e) shows a zoomed region of Figure (d).

- ***RLSA*** is a top-down algorithm which works on binary DIs where white pixels are represented by 0's while black ones by 1's [101, 102]. It processes by converting a binary sequence x into y according to the following rules:
  a) 0's in x are converted to 1's in y if $N_{ad} \leq t_s$, where N_{ad} and t_s are the number of adjacent 0's and the pre-defined threshold.
  b) 1's in x remain unchanged in y.

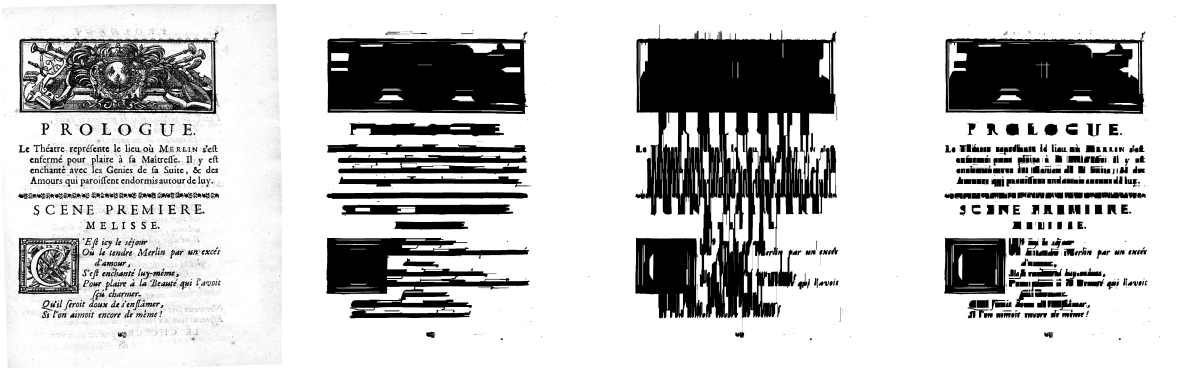
  This step is called a run-length smearing. The RLSA studies the spaces between black pixels in order to link together neighboring black areas, respecting the condition that the black areas are separated by less or equal to t_s white pixels. The RLSA is considered as a special case of morphology-based methods (*i.e.* closing operation with an horizontal structuring element $1 \times t_s$ represents the horizontal smearing). In the RLSA, the run-length smearing step is applied both row-wise (horizontally) to the DI using the horizontal pre-defined threshold t_{sh} and column-wise (vertically) to the DI using the vertical pre-defined threshold t_{sv}. This yields two bitmaps which are afterwards combined using a logical AND operation. The RLSA is considered as a pre-processing task of text/graphic separation, categorization of pre-localized text blocks (e.g. columns, headings, paragraphs, lines, words, notes), optical font recognition (OFR), *etc.* It is very simple to implement and use. It is possible to obtain a separation of text in characters, words, lines or paragraphs, *etc.*, depending on the chosen horizontal and vertical threshold values. Figure 3.11 depicts an illustrative example of the application of the RLSA on a HDI.



(a) Input image (b) Sub-sampled (16×1) image by the threshold reduction operation (c) Seed image (d) Halftone mask image

(e) Input image (f) Sub-sampled (16×1) image by the threshold reduction operation followed by hole-filling operation (g) Seed image (h) Non-text mask image

Figure 3.10.: Snapshots of the original Bloomberg's [118] and the modified text and non-text image segmentation algorithms based on the use of the multi-resolution morphology technique proposed by Bukhari *et al.* [119].



(a) Input image (b) Horizontally smeared image (c) Vertically smeared image (d) Merged image

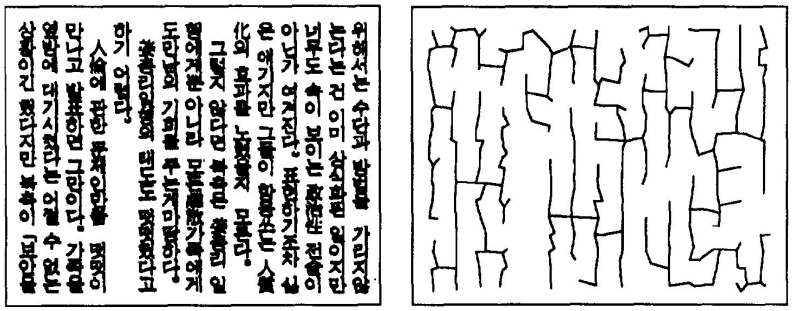
Figure 3.11.: Illustrative example of the application of the RLSA on a HDI. The original image illustrated in Figure (a) is transformed into horizontally smeared (*cf.* Figure (b)) and vertically smeared (*cf.* Figure (c)) images. Then, the RLSA takes the results of applying the logical AND operation between the horizontally and vertically merged images to generate Figure (d).

The original image illustrated in Figure 3.11(a) is transformed into horizontally smeared (*cf.* Figure 3.11(b)) and vertically smeared (*cf.* Figure 3.11(c)) images. Then, the RLSA takes the results of applying the logical AND operation between the horizontally and

vertically merged images to generate Figure 3.11(d), where the horizontal t_{sh} and vertical t_{sv} pre-defined thresholds are equal to $\frac{W}{20}$ and $\frac{H}{25}$, respectively (W and H represent the width and height of a DI, respectively). Similar to the case of the RXYC, the RLSA is well-adapted to DIs with specific layout/structure and when the DI contents are not so varied and it also assumes that the skew has been already corrected. One of the major drawback of the RLSA is that determining the appropriate horizontal and vertical threshold values is a crucial task and high performance in DI segmentation is dependent on these threshold values. As can be seen in Figure 3.11(d), the drop cap is combined with the text block on the right-hand side. In our case, the use of the RLSA is no longer relevant due to the particularities of HDIs such as the presence of irregular spacings (e.g. between characters, words, lines, paragraphs, margins).

3. ***Connected component-based methods***
The CC-based methods are mainly bottom-up approaches which aim to characterize and represent the DI structure among the extracted CCs. There are several CC analysis methods but we just focus and briefly outline the minimum spanning tree (MST), docstrum algorithm and Delaunay triangulation.

   - ***Minimum spanning tree (MST)*** is defined as a the tree structure that connects all the extracted CCs in the form of a graph [120]. The extracted CCs represent the graph nodes and they are connected using vertices or edges. The edges are built between the centroids of the extracted CCs by associating on each edge the computed Euclidean distance between the two CCs and by respecting the minimum sum of distance of edges (*cf.* Figure 3.12). The greedy and Kruskal's algorithms are usually used to built the

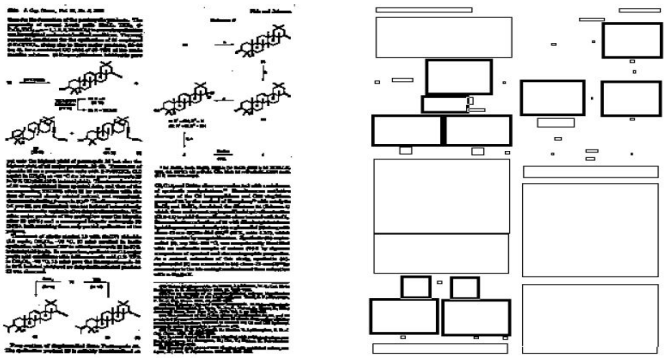


(a) Input image (b) Connected centroids of the extracted CCs by the MST

Figure 3.12.: Illustrative example of the application of the MST on vertical text lines [120].

MSTs [121]. For instance, the Kruskal's algorithm processes by determining the edge having the minimum distance among all possible edges between the CCs and deleting it afterwards from the set of all possible edges. The next step consists in selecting the next edge whose distance is the minimum among the remaining edges with respecting the following rule: the tree structure remains unchangeable after introducing the selected edge. Otherwise, the edge is not recovered and the next one is retrieved. Finally, the resultant tree representing the MST is obtained when there is no more remaining edge and DI components can be extracted as sub-trees of the MST [122]. The DI component extraction is based on formulating an assumption about the distance between the extracted CCs in the same DI component which is supposed smaller than that between the extracted CCs in the different DI components. Thus, by defining a threshold based on the distance between the extracted CCs, the DI segmentation is processed by selecting the edges linking the different DI components and subsequently deleting them. Simon *et al.* [123] proposed a bottom-up method for DI layout analysis based on Kruskal's al-

gorithm in the MST technique (*cf.* Figure 3.13). The proposed method does not require any preliminary separation of the DI components. Nevertheless, it does assume that the text lines of the DI are horizontal. In addition, a skew correction is needed if the DI is scanned with very little skew. Another limitation of the MST technique can be deduced which consists of the possibility to miss edges on the neighboring CCs in the selection and deletion process of edges.



(a) Input image (b) Segmented image

Figure 3.13.: Illustrative example of the application of the MST to extract DI components [123].

- ***Docstrum*** or document spectrum is a bottom-up algorithm proposed by O'Gorman [103] for DI layout analysis based on the NN clustering of the extracted CCs. After a noise removal step, the CCs are classified into two groups: the first group represents the characters of the dominant font size and the second one contains characters in titles and section headings. In this context, a character size ratio factor is pre-defined. Then, the k nearest neighbor (kNN) clustering technique is used, where the nodes represent the extracted CCs, and the edges denote the kNNs from each CC (*cf.* Figure 3.14(b)).

  The proposed algorithm dispenses with the hypothesis that the distance between the extracted CCs in the same DI component is smaller than that between the extracted CCs in the different DI components. In addition, it seeks to prescind from the limits inherent in the MST algorithm by introducing statistical information about the characteristics of the extracted CCs in the stage of the edge deletion and selection.

  Statistical information is deduced by computing a histogram of distance and angle of each CC from its kNNs which is called docstrum (*i.e.* two-dimensional (2-D) plot (d_e, Φ_e), where d_e is the distance between each kNN pair of the extracted CCs and Φ_e is the angle of each edge) (*cf.* Figure 3.14(d)). Based on the computed docstrum, different measures can be calculated such as the peak of the angle histogram (*cf.* Figure 3.14(e)). Clear peaks can be deduced from the docstrum at 0° and ±90° which correspond to angles of edges in text line and between text lines, respectively. Thus, by identifying the highest peak of angle distribution, the dominant skew in the DI is estimated. Two other measures are also deduced for DI segmentation: the within-line and between-line spacing which are estimated from the distance distribution of edges (*cf.* Figure 3.14(f)) close and perpendicular to the estimated skew angle, respectively.

  Firstly, the extracted CCs are merged based on the estimated skew angle and within-line spacing. Then, text blocks are identified by grouping text line and based on the angles of text lines and distances between text lines (*cf.* Figure 3.14(i)). To find page components (*cf.* Figure 3.14(k)), other measures can be determined to adjust parameters of merging the extracted CCs such as the inter-character distance and inter-text line distance. The inter-character distance and inter-text line distance correspond to the

peak with the smallest distance and the peak with the largest distance, respectively. The main disadvantage of the docstrum algorithm is that it supposes that the different deduced measures are globally estimated (*i.e.* several DI components can have different values of measures). Thus, an additional processing is required to handle with the measure space and subsequently to find possible clusters of measures for similar DI components. In addition, setting the appropriate value of k in the kNN clustering technique depends on the layout of the analyzed DI. Thus, if the pre-defined value of k is larger than the appropriate value, spurious edges can be generated in the kNN graph that can significantly falsify the estimation of the measures of within-line and between-line spacing. On the other hand, if the pre-defined value of k is smaller than the appropriate value, the estimation of between-text lines would be inaccurate (*i.e.* edges between text lines will be missing).

- ***Delaunay triangulation*** is a bottom-up method proposed by Kise *et al.* [124] to overcome the limitation of the docstrum algorithm on the defining of the appropriate value of k on the kNN clustering technique. They proposed a particular representation that optimize the construction of edges between the neighboring CCs which is called the Delaunay triangulation (*cf.* Figure 3.15). The generation of the Delaunay triangulation is based on the construction of the Voronoi diagram. The Voronoi diagram $V(P_V)$ is obtained from a point set $P_V = \{p_1, \ldots, p_n\}$ which is called generators and a set of Voronoi regions $V(P) = \{V(p_1), \ldots, (p_n)\}$. A Voronoi region $V(p_i)$ of a point p_i is defined as:

$$V(p_i) = \{p | d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \tag{3.1}$$

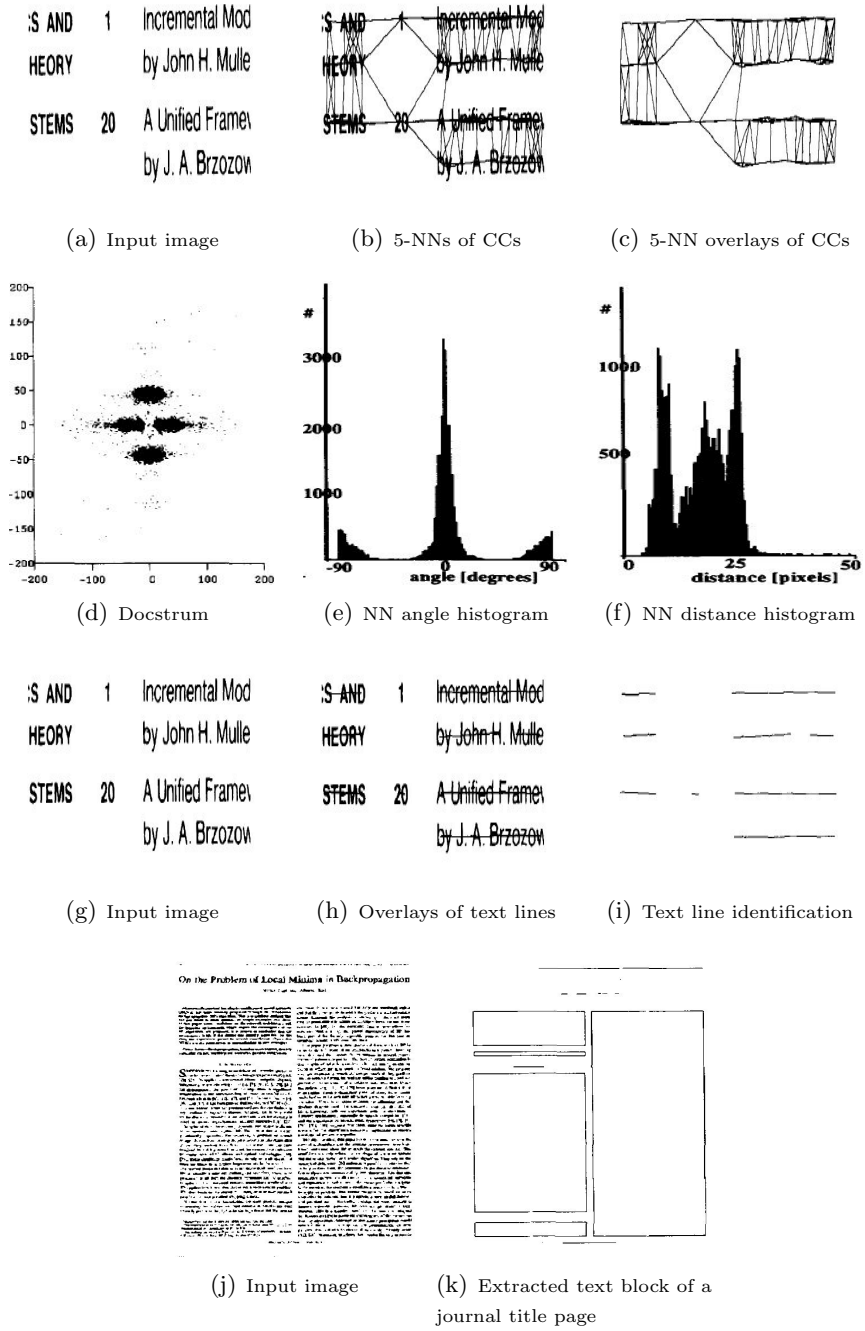
  where $d(p, q)$ represents the distance between points p and q.

  The Voronoi edges are generated by finding the boundaries of Voronoi regions. Thus, the nodes of the Delaunay triangulation represent the point set of the Voronoi diagram, while Delaunay triangulation edges are built between pairs of generators whose Voronoi regions share a Voronoi edge. The generation of the Voronoi diagram and subsequently the Delaunay triangulation is processed by taking the centroids of the extracted CCs as generators in order to obtain the area Voronoi diagram. The area Voronoi diagram is built based on a set of non-overlapping regions deduced from the extracted CCs $G = \{g_1, \ldots, g_n\}$. Indeed, the Voronoi diagram $V(G) = \{V(g_1), \ldots, (g_n)\}$ is constructed, where a Voronoi region is defined as:

$$V(g_i) = \{p | d(p, g_i) \leq d(p, g_j), \forall j \neq i\} \tag{3.2}$$

  where $d(p, g_i) = \min_{q \in g_i} d(p, q)$ represents the distance between a point p and a CC.

  An illustrative example of the application of the area Voronoi diagram for text line extraction is presented in Figure 3.16. The point Voronoi diagram (*cf.* Figure 3.16(c)) is obtained by considering the points of the extracted CCs and taking samples on borders of the extracted CCs as generators (*cf.* Figure 3.17). Thus, the area Voronoi diagram is generated (*cf.* Figure 3.16(d)) after deleting the edges lying between the same CCs. The Delaunay triangulation (*cf.* Figure 3.16(e)) consists of a dual graph of the Voronoi diagram obtained by connecting the different generated regions by the area Voronoi diagram which share the same Voronoi edge respecting the following distance associated with edges: $d(g_i, g_j) = \min_{p_i \in g_i, p_j \in g_j} d(p_i, p_j)$, where p_i and p_j are sampling points of g_i and g_j, respectively. Kise *et al.* [124] proposed a bottom-up DI segmentation algorithm based on the Delaunay triangulation (*cf.* Figure 3.17). They proposed to select edges that connects the different DI components based on the area ratio of the extracted CCs connected by an edge and the angle of an edge to generate the Delaunay triangulation (*cf.* Figure 3.17(b)). Then, by selecting reliable seeds which correspond to parts of text



(a) Input image (b) 5-NNs of CCs (c) 5-NN overlays of CCs

(d) Docstrum (e) NN angle histogram (f) NN distance histogram

(g) Input image (h) Overlays of text lines (i) Text line identification

(j) Input image (k) Extracted text block of a journal title page

Figure 3.14.: Illustrative example of the application of the docstrum algorithm on a portion of a table of contents image [103].

lines (*cf.* Figure 3.17(c)), full text lines can be formed and merged to obtain text blocks based on the text line orientation, length and distance.

3.3.1.2. A short review of classical approaches for historical document image analysis

We have described some representative methods of classical approaches for DIA. For this category of methods, many more algorithms have proposed. The classical methods presented in the literature address various issues and have many limitations in the case of historical DIA.

A well-researched survey dedicated to text line segmentation of HDIs was presented by Likforman-Sulem *et al.* [94]. Most of the existing approaches are based on connectivity features, RXYC [104],

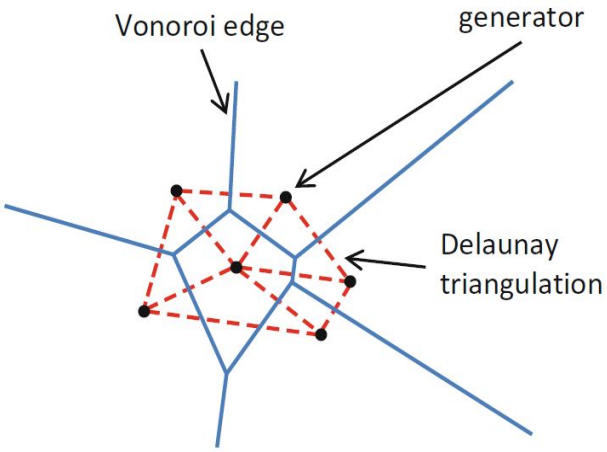
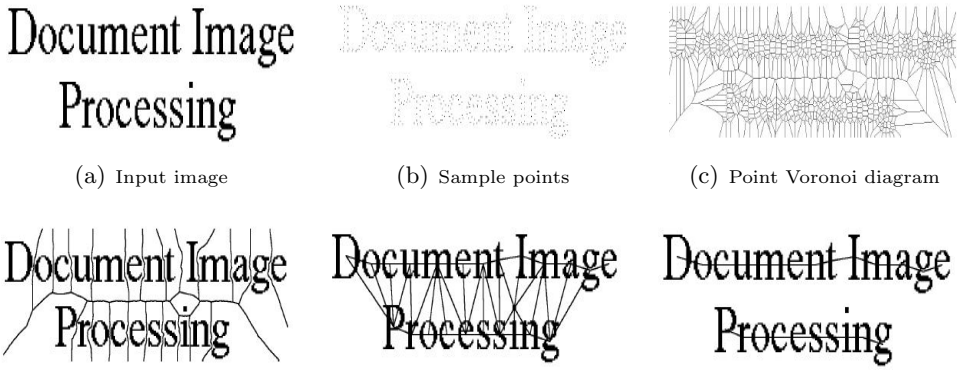


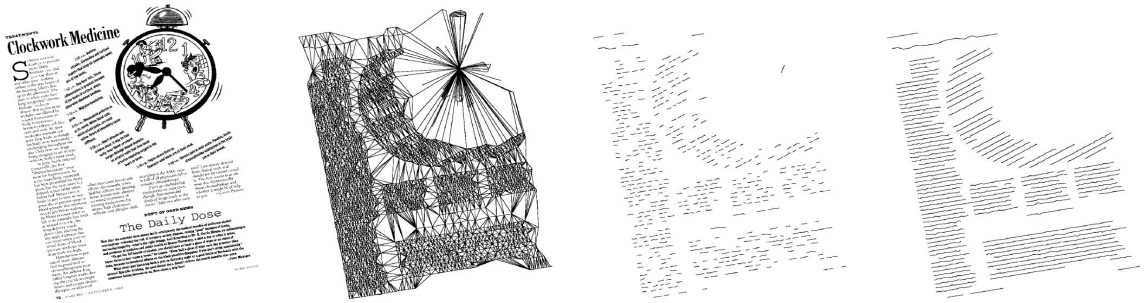
Figure 3.15.: Illustration of a point Voronoi diagram and its corresponding Delaunay triangulation [5].



(a) Input image (b) Sample points (c) Point Voronoi diagram

(d) Area Voronoi diagram (e) Delaunay triangulation (f) Extracted text lines

Figure 3.16.: Illustrative example of the application of the area Voronoi diagram and the Delaunay triangulation for text line extraction [124].



(a) Input image (b) Delaunay triangulation (c) Seeds (d) Extracted text lines

Figure 3.17.: Illustrative example of the Delaunay triangulation for text line extraction from tilted non-rectangular DIs [124].

RLSA [101, 102] and Hough techniques [125] which are suitable for clear lines. These approaches require thresholds to define inter-line or inter-block distances and adjustments for character alignment and line justification. In addition, a pre-processing phase is necessary to remove background noise (superfluous information appearing from the verso) and non-textual regions.

- ***Projection-based methods***
  He and Downton [126] proposed a top-down method based on RXYC approach by alternating projections along the X and Y axes. Few thresholds were defined to estimate inter-line or inter-block distances. The proposed method can only be applied to printed documents (which are assumed to have regular distances) or well-separated handwritten lines.

- ***Smearing-based methods***
  Nikolaou *et al.* [127] proposed adaptive RLSA and skeleton segmentation paths for text line, word and character segmentation of historical and degraded machine-printed DIs. They defined several thresholds and rules in the used segmentation techniques. Even the proposed algorithm worked efficiently for a wide variety of degraded DIs, Nikolaou *et al.* [127] suggested to introduce a dewarping algorithm to improve the overall performance of the proposed approach, particularly for DIs whose text is warped and/or skewed.

- ***Connected component-based methods***
  For instance, Belaïd and Ouwayed [128] proposed a multi-oriented text line extraction approach of ancient Arabic DIs based on image meshing technique, energy distribution of Cohen's class and CC analysis techniques. They defined a few rules depending on the orientations presented in their DIs. Malleron *et al.* [129] proposed a dedicated text line segmentation approach for author's draft handwritings (*i.e.* 19^{th} century handwriting DIs). With formulating a hypothesis that text lines skews can be random and irregular, text line detection is processed by enhancing text line structure using Hough transform and a clustering of CCs to find text line boundaries. Nevertheless, the knowledge of page layout style is necessary to classify corpus DIs and to choose algorithms and decision values for lines and snippets extraction. The proposed algorithm was based on CC analysis, neighborhood-fan computation, corner and borders detection, line orientation estimation, line construction and post-processing.

  Without a given model of the layout for medieval manuscripts, LeBourgeois *et al.* [9] proposed a data-driven layout segmentation approach based on the extracted CCs. Their method required several parameters, estimated thresholds determined by the user and stored in the model, and it also required several pre-processing steps: a binarization step, an image noise reduction filter and the frame removal task based on mathematical morphology [130, 131, 132]. To localize the main body of the text from Arabic manuscripts, they also estimated the average size of text symbols by computing the average size of all CCs. Then, they computed a text probability value for each extracted CC. Finally, they estimated an automatic threshold for each profile (horizontal and vertical) obtained from the entire image. They considered their algorithm to be a useful tool to detect the main body of a text, even for Latin manuscripts, but it did not work with large annotation areas in the margins.

- ***Hybrid methods***
  Gatos *et al.* [133] proposed a segmentation method of historical handwritten documents into text zones and text lines. For text zone detection, vertical rule lines were detected based on using a fuzzy RLSA [134]. On the other hand, vertical white runs and the extracted CCs were afterwards investigated for text line segmentation. In addition, an enhancement of an existing approach based on Hough transform was proposed to analyze vertical connected characters. Thresholds and heuristics were defined for detection of vertical text zones based on vertical rules lines and vertical white runs.

  In the context of the Philectre project, André *et al.* extracted drop caps and text regions from the foreground layer of the analyzed document using edge detection for dark regions (*i.e.* low mean gray level) followed by a thresholding phase that takes into account the local and global adjacent neighboring pixels [135]. Secondly, they used a vertical and horizontal projection phase based on a few thresholds (average height and line spacing) and specified rules for the extraction of columns and lines. Finally, they performed a CC labeling based on

a defined projection interval. This approach was based on a knowledge-acquisition phase to determine the relevant characteristics of a sample set of HDIs.

LeBourgeois and Emptoz [9, 32], as part of the European project DEBORA, analyzed and segmented ancient books using morphology, texture and a bottom-up model. They succeeded in segmenting the physical layout except for some errors which appeared when there were lines of text that were touching, due to a lack of *a priori* knowledge and the highly complex layout of the document. They separated text from non-text regions by combining texture, component shapes and alignments. The recognition of drop caps and strips was based on an *a priori* model designed using information about size, location, surrounding neighborhoods, *etc.* Ramel *et al.* [72] evaluated various traditional methods used for segmentation of historical printed DIs. They highlighted the limits of the traditional methods to segment HDIs. Thus, they proposed a hybrid segmentation algorithm based on CCs for user-driven page layout analysis of historical printed books. The proposed algorithm used two maps: a shape map for foreground information analysis based on CC analysis technique and a background map for white area analysis. Then, the classification of the extracted blocks by using CC analysis technique, was built according to scenarios defined by the user.

3.3.1.3. Discussion

The classical methods used for DIA in the literature are summarized in Table 3.1 by detailing their primitives and strategies and by showing their pros and cons. Nevertheless, the question of finding the best algorithm among them is important. Thus, several research activities have been carried out to answer to this question. For instance, many DI segmentation contests have been performed since 2001 as competitions in biannual ICDAR conferences [136, 137]. These competitions have been performed to compare the performance of classical methods using digitized DIs from commonly occurring publications (e.g. newspapers). Antonacopoulos *et al.* [136, 137] stated that there is still a considerable need to develop robust and versatile algorithms that deal with complex DI layout.

Another research activity has recently emerged which consists in publishing benchmarking and comparative studies of the existing DIA algorithms. This second trend of research activities is of greatest interest to research DIA community. Several works have been tackled the DI layout analysis and particularly DI segmentation, exploiting different rules on the page structure and based on strong *a priori* knowledge. For instance, Mao *et al.* [92] detailed a literature survey of classical DI layout analysis algorithms. They analyzed past works on document physical layout representations and analysis on the one hand, and document logical structure representations and analysis on the other hand. Afterwards, they summarized the limitations of classical approaches. They concluded that formal models for DIs are required to deal with an appropriate level of complexity for a given class of DIs, to estimate model parameters, to analyze and synthesize DIs and to generate synthetic DIs. In addition, they outlined the importance of having relevant physical layout analysis to handle with the logical one. Moreover, the use of a set of objective evaluation criteria is significantly important for quantitative performance evaluation.

Beyond this point, Shafait *et al.* [138] proposed a performance evaluation and benchmarking of six page segmentation algorithms: RXYC [139], RLSA [101, 102], white space analysis [114], constrained text-line finding [110], docstrum [103] and Voronoi [112]. They concluded that the best algorithm must be determined based on the constraints of the analyzed DIs. For instance, the docstrum and Voronoi algorithms are not the best choices when there is no skew and if the DI layout is rectangular or Manhattan.

Moreover, Mao and Kanungo [140] proposed a performance evaluation methodology for evaluating page segmentation algorithms: RXYC [139], docstrum [103] and Voronoi [112]. They concluded that the performance indices of the Voronoi and Docstrum segmentation algorithms are not significantly different from each other. They outlined that the RXYC algorithm is a bad choice if the analyzed DIs have large skew angles and/or large noise blocks. In addition, they stated that the

Voronoi algorithm is a better choice than either the Docstrum or RXYC algorithm for segmentation of DIs with lines separating zones. Moreover, the RXYC algorithm is the best one among the surveyed algorithms in terms of the computational cost (*i.e.* processing time). Nevertheless, it is worth noting that for computational time and memory requirement, the size of the analyzed DIs is considered as an important element to be taken into consideration (e.g. projection and smearing-based algorithms).

An important conclusion of the different surveyed classical DIA algorithms is that they are based on strong *a priori* knowledge such as the repetitiveness of document structure in a corpus (*i.e.* blocks shape, uniformity in horizontal and/or vertical spacings and/or assumptions about textual and graphical characteristics such as font size, *etc.*) There are certain limitations of this family of DIA methods: Firstly, several parameters and thresholds must be adjusted. Secondly, those methods are sensitive to noise and not robust to slanted texts. Other main drawbacks of those approaches are their dependence on the font size, character space, character size, inter-character spacing, document orientation and line and column space, *etc.* Furthermore, the performance of this family of DIA approaches depends on the particular layout and document idiosyncrasies (e.g. Manhattan layouts). Indeed, HDIs do not have strict layout rules. Thus, a real need of segmentation algorithms exists which should be invariant to layout inconsistencies, irregularities, *etc.* Thus, those methods are not well-adapted to degraded and complex layout DIs. Then, for complex and degraded HDIs, it is a difficult task to set empirical rules, domain specific constraints and thresholds. This family of DIA methods are devoted to contemporary DIs [71, 72]. Therefore, based on strong *a priori* knowledge, the classical approaches are not effective for complex and degraded HDIs.

Table 3.1.: Classical DIA methods reviewed by Kise [5].

Ref.	**Tool**	**Primitive or Representation**	**Strategy**	**Advantage / Disadvantage**
A- Foreground-based analysis methods				
1- Projection-based methods				
[104]	RXYC	Projection profile	Top-down	(−)Well-suited to printed DIs (e.g. newspapers) where the document is well-structured and divided into rectangular blocks (−)Not well-adapted to varied, complicated and complex DI layout or if the DIs are skewed or have overlapping layout
[105]	White streams	Projection profile	Top-down	(−)Well-suited to documents that contain rectangular and clearly demarcated blocks (e.g. newspapers, technical documents) (−)Not well-adapted if the DIs have overlapping layout
[115]	Syntactic segmentation	Projection profile	Top-down	(+)Backtracking to correct mistakes (−)Only applied on families of technical documents that share the same layout conventions (e.g. IBM Journal of Research, Development and IEEE Transactions on PAMI)

Table 3.1 – continued from previous page

Ref.	Tool	Primitive or Representation	Strategy	Advantage / Disadvantage
				(−)Not well-adapted if the DIs are skewed or have overlapping layout
[116, 117]	Hough transform	Hough domain representation	Bottom-up	(+)Relatively independent of text font styles, sizes and orientations (+)Adaptive to changes in text characteristics within the DI (−)Not well-adapted if the DIs have overlapping layout (−)Performance dependance on the conformity of the provided constraints and defined heuristics on the analyzed DI characteristics
2- Smearing-based methods				
[101]	RLSA	Smeared pattern	Top-down	(+)Very simple to implement and use (−)Not well-adapted if the DIs have overlapping layout (−)Performance and result dependance on the chosen horizontal and vertical threshold values (−)Use of a post-processing step for text/graphic separation, categorization of pre-localized text blocks, *etc.*
[118, 119]	Morphology	Pixel	Bottom-up	(−)Not well-adapted if the DIs have overlapping layout (−)Definition of the appropriate structural element (−)Skewed DIs can be analyzed with morphology-based methods when the structural element is isotropic (−)Computational burden (−)Use of the multi-resolution morphology to overcome the limitation of the computational burden (−)Use of varied threshold values in the multi-resolution morphology
3- Connected component-based methods				
[103]	Docstrum	kNN	Bottom-up	(+)No assumption concerning the distance between the extracted CCs in the same DI component (+)Introduction of statistical information about the characteristics of the extracted CCs in the stage of the edge deletion and selection

Table 3.1 – continued from previous page

Ref.	Tool	Primitive or Representation	Strategy	Advantage / Disadvantage
				(−)Pre-definition of a character size ratio factor (−)Global estimation of the different measures used in the stage of the edge deletion and selection (−)Requirement for an additional processing to handle with the measure space and subsequently to find possible clusters of measures for similar DI components (−)Dependance of the appropriate value of k in the kNN clustering technique on the layout of the analyzed DI
[120, 122, 123]	MST	MST	Bottom-up	(+)No need of a preliminary separation of the DI components (−)Based on formulating an assumption about the distance between the extracted CCs in the same DI component which is supposed smaller than that between the extracted CCs in the different DI components (−)Assumption that the text lines of the DI are horizontal (−)Requirement for a skew correction step if the DI is scanned with very little skew (−)Possibility to miss edges on the neighboring CCs in the selection and deletion process of edges
[124]	Delaunay triangulation	Delaunay triangulation	Bottom-up	(+)Relatively able to extract text lines independently of layout and skew (+)No need to pre-define the appropriate value of k on the kNN clustering technique (+)Fast (−)Pre-defined parameters for the construction of the Delaunay triangulation (−)Limitations to extract text lines with only a local evidence
B- Background-based analysis methods				
[109, 110]	Shape-directed covers	Maximal empty rectangles	Top-down	(+)Neither prior knowledge of the symbol set nor heuristics (+)Fast and easier to implement than prior methods

Table 3.1 – continued from previous page

Ref.	Tool	Primitive or Representation	Strategy	Advantage / Disadvantage
				(+)Versatile and adapted to multiple-typeface English text and single-typeface Greek, Tibetan, Swedish, chess notation and typeset mathematics (−)Pre-defined aspect ratios and limits for the selection of white rectangles (−)Need of decision tree to train and estimate the probability that a given white space rectangle is part of the page background (−)Need of a deskewing step (−)Only well-adapted if the DIs have Manhattan layout
[111]	White tiles	White tiles	Top-down	(+)Well suited to complex layouts (+)Flexible and fast (+)No need for skew detection and correction (−)Use of a pre-defined threshold to concatenate white runs fo the smearing task (−)Based on formulating an assumption about the required size of printed regions of any given type (text, graphics, line-art, *etc.*) is the largest possible one (−)Is not rotation invariant (−)Determination of merging white tile parameter upon a possible range of skew angle
[112, 113]	Voronoi diagram	Voronoi edges	Top-down	(+)Use of a rotation invariant representation (+)Dynamically adapted to local variations in the size, orientation and distance of components within a DI (+)Well-adapted to obtain the candidates of boundaries of DI components from DIs with non-Manhattan layout and a skew (−)Pre-defined parameters and thresholds for the construction of the Voronoi diagram (−)Based on formulating assumptions about body text regions (*i.e.*, dominant and uniform) (−)Possibility to a over-segmentation of figures, tables and halftones as well as titles with larger fonts, headers and footers with wider inter-word gaps

3.3.2. Texture-based approaches

The ultimate objective of texture-based method is to provide a meaningful image segmentation by extracting textural features and analyzing the produced feature space [141, 142]. By using texture analysis methods, a partition of the analyzed image into regions will be generated. The obtained regions have homogeneous characteristics and similar properties with respect to the extracted features. In this context, texture analysis methods have been classified into three categories [142, 143]:

- ***Region-based approaches*** are used to identify uniform, similar or homogeneous textured regions.
- ***Boundary-based approaches*** are used to analyze the differences in texture in adjacent or neighborhood surrounding regions.
- ***Hybrid approaches*** combine the region and boundary-based algorithms.

It is widely believed that extracting and analyzing texture features on images is still relevant for many applications in image processing and pattern recognition fields [144]. Wechsler [141] stipulated that texture analysis approaches play a fundamental role for many applications in the image processing and patterns recognition fields. Yet, texture has been remained a relevant processing tool for the analysis of many types of images. Texture is considered by Haralick [145] as an important characteristic for the analysis of many kinds of images. In spite of there is not a precise definition of texture, many applications in the areas of biomedical image processing, industrial automation, remote sensing, DIA, *etc.* have benefited of the proposed texture-based algorithms in the literature. A most common definition of texture has been proposed as a set of basic local patterns repeated respecting a periodic arrangement and specific direction over some image region (*cf.* Figure 3.18) [146, 147]. Pratt *et al.* [144] stated that such definition of texture is more appropriate to deterministic kinds of texture (e.g. line arrays, checkerboards, hexagonal tilings). Julesz [148] considered texture as a set of visual and homogeneous characteristics of surface which can be evaluated qualitatively by the human visual system through the visual primitives. Latter, a general definition of texture is given as a measure of the variation in intensity, measuring properties such as smoothness, coarseness and regularity [149].

Figure 3.18.: Illustrative example of four kinds of texture.

3.3.2.1. Categories of texture-based approaches in image analysis

By referring to the formal definitions of texture, two different texture-based approaches which are called statistical and structural, give rise due to their particularities of texture (*i.e.* stochastic

or repetitive structure of texture) [141]. Julesz [10] characterized the statistical and structural approaches as perceptual and cognitive, respectively. Pickett [146] considered the statistical approaches as "impressionistic" ones since they help to characterize texture as being coarse or fine, while the structural approaches are considered as "delibrate" ones because they involve arrangement analysis and they are more complicated. Haralick [145] surveyed the statistical and structural approaches investigated in the image processing field. The texture features extracted from the auto-correlation function, optical transforms, digital transforms, textural edginess, structuring element, gray tone co-occurrence, run-lengths and auto-regressive models are considered as statistical approaches. Some structural approaches based on more complex primitives than gray tone (e.g. Zucker's model [150]) was presented in this survey. Haralick [145] concluded that the statistical techniques can be applied to the structural primitives to generate some structural-statistical generalizations.

Different categorizations and classifications of texture-based methods have been presented in the literature. Toyoda and Hasegawa [151] classified texture-based approaches into two kinds: local (e.g. Gaussian Markov random fields (GMRF) [152], local binary patterns (LBP) [153]) and frequency methods (e.g. wavelet transform [154], Gabor filters (GFs) [155]). Feddaoui and Hamrouni [156] categorized texture analysis methods into three approaches: structural, statistical and spatio-frequency. Zhang and Tan [157], and Reed and DuBuf [142] classified the invariant texture texture analysis methods into three categories:

1. ***Feature-based methods***
   The feature-based methods are used by extracting textural characteristics which are relatively constant in homogeneous and similar content regions. Operator-based (e.g. texture energy measures formulated by Laws [158]), statistic-based (e.g. gray-level co-occurrence matrix (GLCM), Tamura [159]) and transform-domain (e.g. power spectrum peaks, shape of the power spectrum [160]) are considered as the derivatives of feature-based methods.

2. ***Model-based methods***
   The model-based methods have been introduced to model and characterize texture by using the coefficients of probability model or linear combination of a set of basis functions. The fractal models, stochastic models (e.g. Markov model) and decision-theoretic techniques (e.g. simultaneous auto-regressive model (SAR) [161]) are few kinds of model-based methods. Wold-like model [162], multi-channel GFs, steerable pyramid (SP) [163] and wavelet transform are considered as model-based methods by Zhang and Tan [157] while spatial/spatial-frequency techniques as sub-class of model-based methods by Reed and DuBuf [142].

3. ***Structural methods***
   The structural methods consider texture as many textural elements which are called texels, arranged according spatial organization rules (e.g. perimeter and compactness [164], invariant histogram [165], topological texture descriptors such as Hough transform [166], morphological decomposition [167]).

3.3.2.2. Categories of texture-based approaches in document image analysis

Okun and Pietikäinen [6] classified texture-based layout analysis approaches into two categories: "Group 1" and "Group 2". A summary table of these reviewed texture-based methods, describing briefly their algorithms, parameters, inputs and outputs, and showing their pros and cons, are presented in Table 3.3. The first class of texture-based methods "Group 1" is firstly processed by extracting document regions using smearing techniques in most approaches. Then, each region is classified according to the extracted textural features. This category of methods has the disadvantage that its performance depends on the quality of the region extraction phase. The second class of methods "Group 2" is processed by extracting textural features from a given analysis window of $(M \times M)$ pixel size, where $M < \max(W, H)$, and W and H are the width and height of the

analyzed DI, respectively. The analysis window technique can be performed by two ways, pixel-wise or block-wise [168, 143]. The first one which is called pixel-wise, is applied by considering many overlapping sliding windows in such a way that a DI is analyzed pixel by pixel so that each pixel had its proper class label while the second way which is called block-wise, is performed in such a way as the analyzed document is partitioned into many non-overlapping windows so that each window had its proper class label. The use of the overlapping windows ensures an accurate localization of region boundaries, although this is paid for with longer computation time. Then, the document regions can be obtained by applying a window/pixel classification and relaxation/post-processing step. A post-processing task is often essential for this category of texture-based methods to merge pixels or windows into larger regions or blocks. Since the "Group 2" of texture-based methods has been considered as a local processing technique, Okun and Pietikäinen [6] stipulated that this class of methods is more robust to different document layouts and/or DI skew than the "Group 1". They pointed out that the main problems of texture-based methods for document layout analysis are closely linked to the extraction of small text components inside graphics and the big character detection. The big characters are often localized in document titles, document headings or drop caps while text contents which are embedded in graphics, are typically located in charts or plots. Secondly, they also underlined an important issue of using texture-based methods which consists of their quite high computational complexity. Moreover, their processing time depends on the image sizes (and resolution) due to the use of pixel-based computation, large DI size and high complexity of texture analysis approaches.

On the other side, Cote and Albu [3] classified the most widely used texture-based methods by the DIA community into two categories, the statistical and spectral methods. The statistical approaches investigate the spatial distribution of gray-levels within a region of interest while the spectral ones describe texture by frequency descriptors obtained by computing the response of an image to a given filter bank. Cote and Albu [3] reported that most spectral approaches do not require nor a binarization step neither a prior region segmentation. On the other side, the statistical approaches are categorized into two classes, those proposed for textural description of pre-segmented regions of DIs and other methods for document segmentation or pre-processing. The former approaches require binarized images in most cases and work on pre-segmented homogeneous blocks. For example, Wang and Srihari [169] extracted statistical features based on black-white pair run-lengths and black-white-black combination run-lengths from pre-segmented homogeneous blocks in binarized newspaper images. Chetverikov *et al.* [170] introduced the feature-based interaction map (FBIM) for classifying already partitioned homogeneous zones in text or non-text. Eglin and Gagneux [171] extracted several statistical features (e.g. entropy, directional compactness, visibility) for categorizing pre-localized text blocks into headings, paragraphs, notes (head-notes and foot-notes) and abstracts, *etc.* from scientific DIs. They proved that textural properties are appropriate to typography characterization of text fonts. For characterization of functional blocks in DIs (*i.e.* labeling of pre-segmented text lines), Allier *et al.* [172] proposed a texture-based approach by combining several features (e.g. black/white transitions, entropy, compactness index in a given direction, tiling, histogram entropy, eccentricity). The latter approaches for document segmentation or pre-processing do not need prior region segmentation or *a priori* knowledge about layout or content. For example, Payne *et al.* [173] [174] used the texture co-occurrence spectrum technique (TCS) to classify regions into text, image, *etc.* from binarized DIs. For text/textured-background separation, Chen used the sequential directional energy of pixels as texture features by applying the coarse-to-fine segmentation technique (*i.e.* from coarse block classification into text, background or boundary regions to pixel-level segmentation). For segmenting DI contents into text, graph, table and picture, Kim and Kim [175] analyzed six standard GLCM features. Journet *et al.* [1] extracted three auto-correlation features which were derived from the rose of directions and two frequency attributes by using a multi-scale analysis for classifying HDI pixels into text, graphics and background. The first frequency descriptor computes the ink/paper transitions obtained by performing the average per-line sum of the difference between pixel intensity value and its left

neighbor. The frequency second attribute calculates the white spaces obtained by performing the RXYC algorithm and computing the mean of the average per-line and per-column sums of pixel intensities over an analyzed area. Then, by using the clustering large applications (CLARA) [176], an unsupervised clustering algorithm, the extracted texture descriptors were clustered and pixels were separated into different content clusters (*cf.* Figure 3.19). 83% and 92% mean good classification rates were noted for the graphical and text pixels, respectively with 180 minutes in total per document as time required to process a page (feature extraction and pixel-clustering tasks).

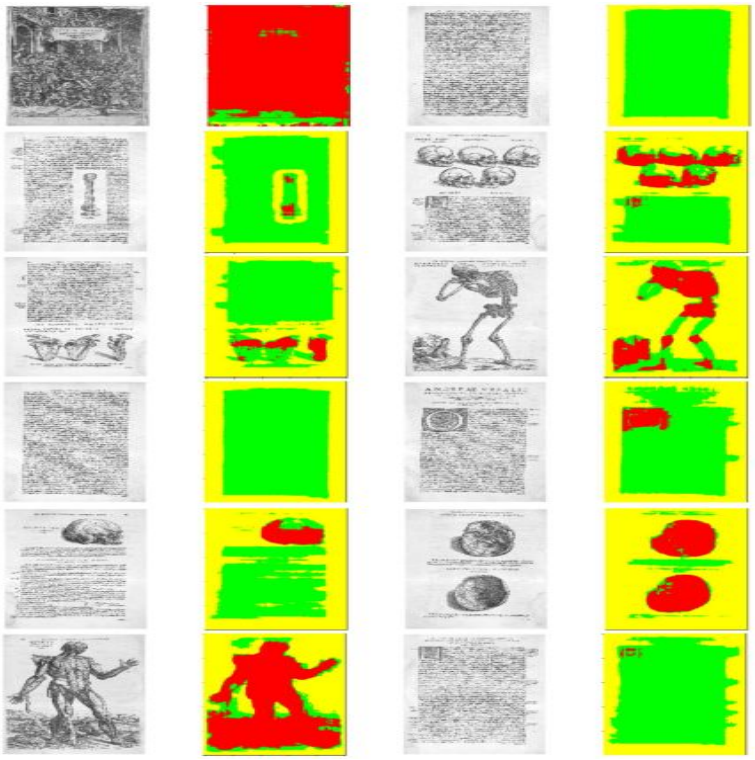


Figure 3.19.: Result examples of Journet *et al.*'s [1] texture-based approach for pixel-labeling of historical book content.

In our view, texture feature extraction and analysis methods may be categorized into five classes according to the properties or characteristics of the extracted textural features [177, 178]:

1. ***Statistical methods***
   The statistical methods are used to analyze the spatial distribution of gray levels by computing local indices in the image and deriving a set of statistics from the distribution of the local features. The statistical methods have the advantage of being simple to implement and their effectiveness is proved. The auto-correlation function [145, 179], GLCM [180] and gray-level run-length matrix (GLRLM) [181] are three standard statistical methods. By computing some indices on the GLCM [182], the texture regularity and repetitiveness are characterized. Caponetti *et al.* [183] proposed a document page segmentation method using a neuro-fuzzy methodology on statistical features. Another approach was proposed by Journet *et al.* [1] that is devoted to HDI segmentation based on extracting and analyzing texture features. The extracted texture features are based on frequencies and the auto-correlation function. This method gives good information on the principal orientations and periodicities of the texture allowing to characterize the content of images without any assumption on the image structure or properties. Although their results are promising, their algorithm is computationally expensive because it is carried out for each pixel and the size of the analysis window is a critical parameter that is difficult to determine. Uttama *et al.* [29] introduced a drop cap

segmentation method based a combination of different texture analysis approaches (GLCM [180], auto-correlation function [1], *etc.*)

2. ***Geometric methods***
   The geometric methods are used to describe intricate patterns and to retrieve and describe texture primitives by characterizing the notion of a texton. Texture primitives may be extracted using a difference-of-Gaussian filter, for example [184]. Those methods attempt to characterize the primitives and find rules governing their spatial organization. Among the classics of geometric methods, moment-based texture segmentation is one of the well-known methods. Anyway, moment-based texture segmentation is not sufficient to discriminate all types of texture and the algorithm needs a non-linear transformation of the images [185].

3. ***Model-based methods***
   The model-based methods are used to compute a parametric generative model based on the intensity distribution of texture primitives. A widely used class of the model-based methods are the probabilistic models. The conditional random fields (CRF) [186], Markov random fields (MRF) [48], Gaussian Markov random fields (GMRF) [152], fractals [187] and LBP [153], *etc.* are the most commonly used tools based on probabilistic models. This category of texture-based segmentation methods is complex to implement. There are many difficulties in the learning phase and a long computation time is required. The MRF are perfectly adapted to DIs with high variability in terms of the layout and the quality of the scanned document which yields good performance in handwritten DIs. However, the MRF are not robust since the learning phase is only valid for one type of document at a time. The fractal dimensions compute measures of texture roughness and repeatability of a pattern. They are considered as a useful tool for image segmentation when the image characteristics tend to be predictable and repetitive and in which the objects to segment tend to be irregular or different from the background.

4. ***Spectral methods***
   The spectral methods are used to investigate the overall frequency content of an analyzed image. The most widely used spectral methods in indexing and segmentation of natural images: GFs [188, 189], Fourier transform [190] and wavelet transform [154, 190]. For instance, the unsupervised texture segmentation algorithm [189] is used to segment an input image into regions of homogeneous texture based on a bank of GFs. GFs have the advantage of reducing the computational complexity and are suitable for document texture analysis. One of the limitations of such an algorithm based on a fixed set of GFs is that many parameters must be fixed [191]. Another frequency algorithm combined the wavelet and the Fourier transform to index image databases [190]. Although the proposed algorithm is faster and more robust than the separate use of the discrete Fourier and wavelet transforms, the computation time is directly dependent on the level of wavelet decomposition. Qiao *et al.* [192] combined the kernel-based methods and a Gabor wavelet to segment DIs scanned from popular newspapers and journals. They confirmed that the multi-scale analysis of an image is ensured and the multi-orientation properties of an image is deduced, but the effectiveness and computational complexity of the proposed algorithm is no longer preserved and a proper post-processing is needed to improve the segmentation result. Another frequency method was proposed to extract three classes (text, background and graphics) from postal images based on six features derived from wavelet transforms [193]. Even if the proposed algorithm has a good recognition rate, one of the features must be adjusted manually and the efficiency (computation time) of the algorithm is limited.

5. ***Hybrid methods***
   The hybrid methods combine different kinds of texture features and other types of descriptors

(e.g. shape, color, topological or spatial descriptors) to address a general issue in image segmentation and analysis.

3.3.2.3. A short review of texture-based approaches for document image analysis

One of the pioneering texture-based work in DIA is a text zone localization method proposed by Jain and Bhattacharjee [189]. This approach considers the text in a document as a textured region, while the non-text contents (e.g. blank spaces, graphics, pictures) are considered as regions with different textures. The use of texture is not limited to text/non-text region separation, but it is extended to font characterization using geometric descriptors [101, 102], statistical features [194] or generic techniques [170]. Another application of texture in DIA was proposed for skew angle detection [195]. In context of the characterization and classification of printed text, Eglin *et al.* [194] considered text regions as a set of little symbols or graphics repeated according to a specific spatial organization which generates a "macroscopic" impression of texture. They defined a text character as an elementary entity of texture. Thus, the character disposition, frequency, font and language in text regions represent the visual characteristics of texture. Statistical features, such as the entropy and compactness, histogram of density variation were extracted to provide a characteristic signature of text. Allier *et al.* [172] adapted the definition of texture as a set of properties such as fineness, coarseness, smoothness, *etc.* which can be used for physical layout segmentation of DIs. They considered texture as a suitable measure for the analysis of the DI physical layout and its block content. For characterization of functional blocks in DIs (*i.e.* labeling of pre-segmented text lines), various texture-based methods (e.g. black/white transitions, entropy, compactness index in a given direction, tiling, histogram entropy, eccentricity) were combined.

Okun and Pietikäinen [6] assumed that text regions have different texture features from non-text ones. Indeed, text areas contain text lines sharing similar characteristics (e.g. approximately similar orientation, inter-character, inter-line spacings). This means that text regions are considered as regular and periodic textures while non-text ones are characterized by irregular textural properties. Thus, in this study, three assumptions are made to ensure a differentiation between various text fonts and numerous types of graphics [10, 189, 194]. First, textual regions in a digitized DI are considered as textured areas, while its non-text content is considered as regions with different textures. Secondly, text with a different font is also distinguishable. Finally, different types of graphics can be perceived as different textures (e.g. drop cap, embellishment, frame, illumination, engraving, stamp, sketch).

Latter, a variety of approaches for characterizing image texture have been investigated in many fields of DIA. They have been used for many DIA applications such image binarization [196], character recognition [197, 198], script and language identification [199, 200], writer identification and verification [201, 202, 203], handwriting classification [30], font recognition [204, 205], printer identification [206, 207], watermarking [208], geometric rectification of warped DIs [209, 210], generating paper texture [211], text/non-text separation in a DI [2], line separation [134], DI segmentation [173, 212, 192], document classification and retrieval [213, 214, 215], *etc.* For instance, Nourbakhsh *et al.* [2] proposed an automatic method for separating text from non-text elements in a complex gray-scale DI (*cf.* Figure 3.20).

The texture-based methods used with HDIs in the literature are summarized in Table 3.2 by detailing their algorithms, parameters, inputs and outputs, and showing their pros and cons.

3.3.2.4. Discussion

Several limitations of statistical and spectral texture-based approaches were presented in [3]. Firstly, most approaches reported in the literature rely on binarized images and require prior region segmentation (*i.e.* they work on pre-segmented homogeneous blocks) or background/foreground separation. Secondly, most approaches assume rectangular shapes for all document elements. Thus, using a texture-pixel-based approach is considered as an interesting alternative to be adapted to

(a) Original image (b) Output image

Figure 3.20.: Result examples of a texture-based approach for pixel-labeling of HDIs proposed by Chen *et al.* [4].

shape flexibility [1, 216]. Then, few spectral and statistical approaches methods have been conducted in the literature for processing the whole digitized DI since the most existing approaches focus on a specific document element, for example the characterization or analysis of graphic images such as drop caps [29] or a particular segmentation or classification task such as text localization [217]. Finally, most texture-based works in DIA reported mainly visual or qualitative results by using statistical or spectral approaches. Cote and Albu [3] confirmed that conducting qualitative and quantitative evaluations is essential for analyzing texture-based approaches used in DIA.

The two most encountered spectral methods in literature are the multi-channel GFs [218] and wavelets [217]. Jain *et al.* [188] used a bank of 20 GFs for text/graphic separation on scanned newspaper images. For classifying large blocks into background, photograph, text and graphics, Li and Gray [219] investigated two textural features extracted from the distribution of wavelet coefficients. Nevertheless, the research DIA community is continuing to investigate new texture descriptors. Recently, Cote and Albu [3] proposed a low-dimensional texture-based feature descriptor that explored the response sparseness of the input DI to the Leung-Malik filter bank based on the multi-scale and contextual techniques [220]. The Leung-Malik filter bank includes many types of filters such as Gaussian derivative, Gaussian and Laplacien of Gaussian filters. The sparseness has been introduced to reduce the dimensionality of the analyzed textural vectors. The proposed approach was used to classify individual pixels of color business DIs into four fundamental classes (text, image, graphics and background) with 83.36% overall pixel classification accuracy (*cf.* Figure 3.21) and 82 minutes in total per document as time required to process a page (feature extraction and pixel classification tasks).

It is worth noting that finding the best texture-based algorithm is quite hard to address a general issue in DIA. Hence, many researchers have addressed the issues of combining different kinds of texture features to perform a particular DIA task. For instance, Qiao *et al.* [192] combined the Gabor wavelets and kernel-based methods for DI segmentation. Benjlaiel *et al.* [221] proposed to combine Gabor, Fourier and invariant moment features to analyze principal orientations in the annotation region for multi-oriented handwritten annotations extraction from scanned DIs. Eglin *et al.* [30] used the results of the auto-correlation features in order to compute the Gabor descriptors for handwriting classification in ancient manuscripts. Said *et al.* [199] presented a global method for handwriting identification based on the use of GFs and GLCMs. Shahabi and Rahmati [202] proposed a method for writer identification of handwritten DIs by combining the Gabor and co-occurrence features.

On the other hand, combining texture features with other kinds of features (e.g. shape, color, topological descriptors) has been demonstrated significantly relevant. For instance, Pardeshi *et al.* [222] extracted directional multi-resolution and spatial information by combining different kinds of features for automatic handwritten Indian scripts identification. They extracted several fea-

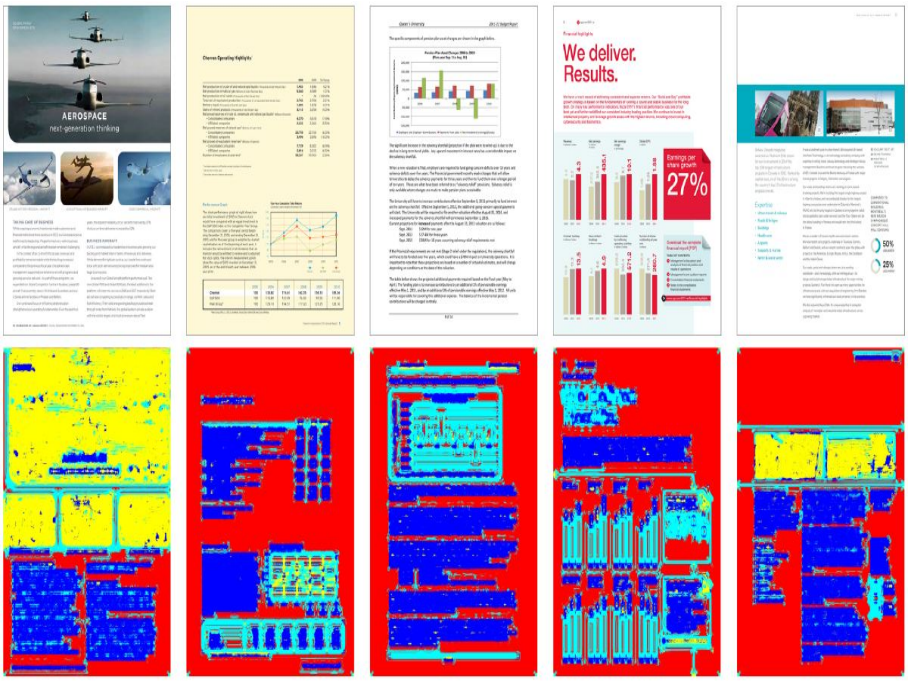
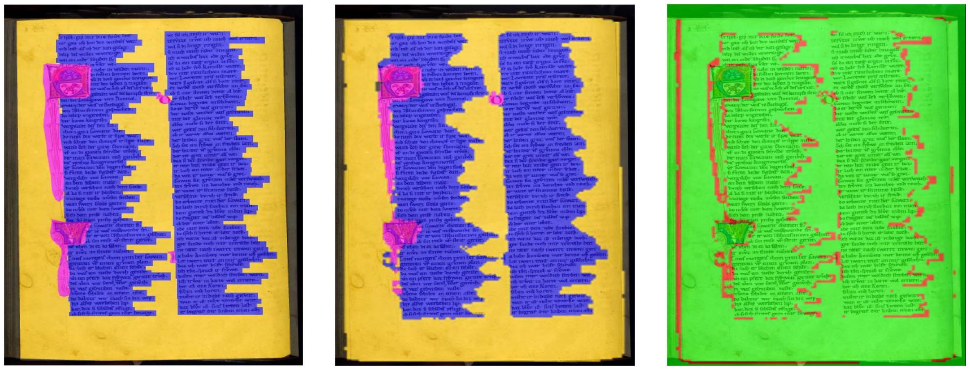


Figure 3.21.: Result examples of a texture-based approach for pixel classification of business DIs proposed by Cote and Albu [3].

tures from the Radon transform, discrete wavelet transform, statistical filter and discrete cosine transform. Maximum accuracies of 98% and 96% were achieved for bi-script and tri-scipt, respectively. Seuret *et al.* [223] proposed a method for discriminating printed content from handwritten annotations at pixel level. They extracted from the foreground pixels and their neighbors several features (mean luminosity, luminosity variance, smoothness, gradient density, arithmetic operators, shannon's entropy, histogram moments, edge detectors, GLCM, side histogram and run-length). Chen *et al.* [4] proposed a physical structure detection method for historical handwritten DIs by classifying and labeling each pixel as periphery, background, text block or decoration. Without any assumption of specific topologies and shapes, coordinates, color and texture features (e.g. color variance, smoothness, Laplacian, LBP, Gabor dominant orientation histogram (GDOH)) were used for classification. A 96.10% of mean accuracy was achieved (*cf.* Figure 3.22).



(a) Ground-truth (b) Segmentation results (c) Pixel-labeling errors

Figure 3.22.: Result examples of a texture-based approach for pixel-labeling of HDIs proposed by Chen *et al.* [4].

Nevertheless, a feature selection step is often required to select relevant features and remove

redundant ones. For example, Wei *et al.* [58] proposed a hybrid feature selection method for historical DIA by using adapted greedy forward selection and genetic selection in a cascading way. They concluded that the proposed feature selection method selected significantly less features and lower error rates (*i.e.* 7.97% of mean error rate was noted) were obtained than in the case of using all features. In addition, they noted that some texture features (e.g. gradient, Laplacian and LBP) were frequently selected. Moreover, Tao *et al.* [224] presented a feature selection based on the dimension reduction technique which is called sparse discriminative information preservation (SDIP) for Chinese character font categorization, after applying the LBP operator.

3.4. Conclusion

Antonacopoulos *et al.* [39] pointed out the significant need for robust and accurate DIA methods that deal with the idiosyncrasies of HDIs. Thus, since 2011 and in the context of ICDAR and HIP, many competitions (e.g. historical document layout analysis, HNLA, HBR) have been organized for the purpose of providing an objective comparative evaluation of the different submitted methods [39, 225, 226]. This kind of competitions has other objectives such as analyzing the performance of each submitted method on a representative dataset in different scenarios (from the scenario of segmenting, labeling and recognizing regions to the text localization and recognition scenario). The submitted methods are based on CC analysis, horizontal and vertical separators for text line and region extraction. There are certain limitations of these methods: firstly, several restrictions on the extracted CC properties (e.g. extracted CCs are filtered out according to the size and spacing values). Secondly, those methods are sensitive to severe page curl or arbitrary warping. Finally, these methods require a number of pre-processing tasks (e.g. local or global binarization algorithm, skew correction step, page border removal phase).

In addition, Crasson and Fekete [227] highlighted the real need for automatic processing of digitized HDIs (HDI layout analysis and text/non-text separation) to facilitate the analysis and navigation in the corpus of ancient manuscripts. Moreover, Kise [5] stated that the analysis of pages with constrained layouts (e.g. rectangular, Manhattan) and clean DIs has almost been solved while historical DIA is still an open problem due to their particularities (e.g. noise and degradation, presence of handwriting, overlapping layouts, great variability of page layout). He also precised that the most relevant methods used to analyze pages with overlapping or unconstrained layouts are based on signal properties of page components by investigating texture-based features and techniques. Hence, texture-based methods address the challenges of the existing state-of-the-art ones. The use of texture-based methods for DIA has been shown to be effective with skewed and degraded images [228]. Thus, in this work we explore various aspects of the texture features in HDIs to assist the analysis of DIs by characterizing a DI layout through a set of homogeneous regions. Given that there are significant degradations and no hypothesis concerning the layout, the graphical properties or typographical parameters of the analyzed HDI, the use of texture analysis techniques for HDI has become an appropriate choice.

Table 3.2.: Texture-based methods used with HDIs in the literature.

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
1- Statistical methods							
[89, 229]	Auto-correlation function	-Entire color high resolution digitized images (manuscripts and printed) -Italian	Illustrations	-Size of the analysis squared block -Size of the squared structural element -Factor of the down-scale -SVM kernel parameters	Automatic layout analysis and content enrichment of DHBs and illustration segmentation in HDIs using local auto-correlation features. First, the RXYC algorithm with a pre-processing phase of binarization and morphological closure were performed to extract the main regions from the page and ensure the geometric layout analysis. Then, each region was divided in small squared blocks, and the auto-correlation features were computed on each block. The lauto-correlation features were deduced from a directional histogram obtained from the projections of the auto-correlation matrix along the vertical and horizontal directions in order to identify the repeating pattern of the texture. To segment text and illustration of digitized old documents, a supervised learning approach using a texture feature based on the auto-correlation function was performed on the extracted regions. The proposed approach aimed at detecting the repeating patterns of text regions and differentiate them from pictorial elements. Text, images and their associated captions were extracted using a SVM classifier trained on the extracted texture features. A 308-dimensional feature vector for each block was constructed. To enrich the manuscripts with new related contents, extracted images and keywords contained in their captions were used to retrieve similar images from the Web.	-Effective on several historical datasets -Outperformed the state-of-the-art methods in presence of challenging documents with a large variety of pictorial elements	-Not parameter-free -Requirement for a training task -Need for a pre-processing task for geometric layout analysis (the main regions were extracted from the page using the RXYC algorithm)

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
[169]	Run-lengths	Binary image digitized at 100 or 200 dpi	Rectangular regions classified as titles, paragraphs, line drawings or pictures	-Smearing white and black thresholds -Three parameters of merging process -Column gap width -Weight and threshold vector used for the classification algorithm -Ratio parameter used for the classification algorithm	Statistical features based on black-white pair run-lengths and black-white-black combination run-lengths were extracted from pre-segmented homogeneous blocks in newspaper images. RLSA and XY-CUT algorithms were used to segment a document into homogeneous regions as pre-processing task for extracting texture features.	Quite fast	-Based on strong *a priori* knowledge -Not parameter-free -Use of RLSA and XY-CUT algorithms to segment a document as a pre-processing step
[230]	-Auto-correlation function -Rose of directions -Multi-scale analysis	-Entire binary, gray-scale or color pages (manuscripts) -German and Latin	Identified text regions in ancient manuscripts	-Number of SVM classes -SVM kernel parameters -Sizes of the analysis windows	Text recognition in ancient manuscripts was carried out by firstly extracting the auto-correlation features using a multi-scale technique. Then, the classification task was performed with SVM. Three auto-correlation features proposed by Journet *et al.* [1] were extracted by applying three scales by means of overlapping sliding windows. Shifted copies of the proposed textural features proposed by Journet *et al.* [1] such that the main orientation is at 0° has been introduced to ensure the comparison of different roses of directions and the invariance to skewed text lines. Two logical classes were defined: a class for regular text and another class containing all other regions such as background, initials or headlines.	-Tolerant towards varying character sizes, skewed text blocks and faded-out ink -Working on non-rectangular and unstructured layouts although rectangular sliding analysis windows are selected -Applied to pages that are different in writing style and line spacing -High performance obtained for the new introduced auto-correlation based features compared to the ones proposed by Journet *et al.* [1] and local projection profiles algorithm	Use of a learning phase
[231]	Statistical indexes from pixel mask	-Binary drop caps (printed) -French and Latin	Classified drop caps	-Size of the analysis mask -Number of clusters	Drop cap indexing by using four classes recognition system. A quantization task was firstly applied on the original image to obtain three gray-levels image.	High recognition rate for most of styles and especially for the black style of the analyzed drop caps	Dimension of the representation space is quite high (*i.e.* it is equal to 729)

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
					Then, three textural descriptors (rank, frequency and tf-idf associated with frequency) were extracted by scanning an image using a 2×3 image mask and used thereafter to compare drop caps with the nearest neighbor classifier.		
2- Geometric methods							
[47, 48]	Bi-scale feature vector based on the pixel density measurement and HMM	-Entire binary pages or parts of pages (manuscript drafts set of *"Madame Bovary"*) -French	Words (or parts of words) and deletions	-Parameters of the Gaussian mixtures -Number of the connected neighbors -Size of the analysis region -Interline space -Number of the states to separate words	For the segmentation of Flaubert's manuscripts into their elementary parts (text lines, erasures, punctuation marks, interlinear annotations, marginal annotations, *etc.*), relevant signatures were computed. These signatures were generated by constructing bi-scale feature vectors based on the pixel density measurement and HMM.	-Good results for separating words (or parts of words) and deletions compared to those obtained by using the approaches based on the CC level analysis -Pixel-level approach which ensures segmenting different page parts which are connected together -Possibility to extract text lines or other objects of higher level (e.g. text blocks) by applying few label merging rules	-Adapting to manuscripts which are characterized by some typical layout rules (e.g. an important text body occupying $\frac{2}{3}$ of the page, the presence of erasures and a marginal area with some text annotations) -Working on binary images -Parameters of the Gaussian mixtures when modeling and learning the probability densities through the manually labeled images using the expectation–maximization (EM) algorithm -Size of the analysis region (5×5 pixels) was set for the extraction of pixel densities to ensure the extraction of small page elements such as the diacritics -Need to know the interline spaces to ensure good segmentation of text lines and text block detection -Only qualitative results obtained on few images of full page of handwriting or parts of pages from the Bovary database were presented -Dependency of the quality of segmentation on the merging strategy and choice of the extracted features
[232]	Logarithmic base of histogram entropy	-Entire color typed pages (Nabuco's bequest) digitized at 200 dpi -Latin	High quality monochromatic DIs	Two multiplicative factors for entropy rule definition	An entropy-based segmentation algorithm was proposed for back-to-front noise reduction of documents written on both sides. The segmentation process was used to generate high quality gray-scale or monochromatic images for improving OCR performance.	-With the use of a fidelity index, the segmented images can be evaluated quantitatively -Experiments were conducted on a large corpus of HDIs (*i.e.* 500 samples from Nabuco's bequest)	Empirical definition of two multiplicative factors for entropy ranges and rules

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
					An image quality measure which is called a fidelity index, was defined to choose the best logarithmic base of histogram entropy when generating the different threshold values.		
[233]	Histogram entropy	-Entire color typed pages (Nabuco's bequest) digitized at 200 dpi -Latin	High quality monochromatic DIs	Two multiplicative factors for entropy rule definition	An entropy-based segmentation algorithm was proposed for back-to-front noise reduction of documents written on both sides. The segmentation process was used to generate high quality gray-scale or monochromatic images for ensuring an automatic transcription of HDIs. By analyzing the most frequent color belonging to the image background, an initial threshold value was set to compute the histogram entropy of the image and determine the limit values of the two multiplicative factors for entropy rule definition.	Promising results in terms of the OCR hit rates and visual inspection of monochromatic images quality	-Empirical definition of two multiplicative factors for entropy ranges and rules -Experiments were only conducted on a set of 40 samples from Nabuco's bequest
3- Model-based methods							
[51]	Meyer-based decomposition	-Binary, gray-scale or color drop caps (printed) digitized at 300 dpi -French and Latin	Letters extracted from drop caps	-Parameters of the Meyer decomposition -Size of the mask for Zipf modeling -Number of selected gray-levels for the Zipf law	Decomposition of the information of drop caps into several layers (*i.e.* segmenting the letter and the elements from its background) to characterize them by using a relevant signature. The extraction of the letter was based on the Meyer decomposition (layer segmentation) and Zipf modeling. A Meyer decomposition was used to filter out the noise and to extract the spatial frequencies of drop cap images, to segment them into separate layers (shape layer, texture layer and noise layer). Then, the Zipf law on the gray-levels of the shape layer ensured the detection of large homogeneous areas which correspond to the letter.	-Automatically adaptable to the constraint of the color of drop cap letter (black or white) -Robust toward noise variations -Reduction of dimensionality of textural vector by using the Zipf law	-Requirement for physical segmentation at a drop cap level as a pre-processing step -Failed in the cases of very degraded images, where the letter is composed of many CCs

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
[234]	LBP operator	-Machine printed document pages (archives of Portuguese HDIs) -Portuguese	Labeled regions: background (white or black), text and graphics	-Number of models -Number of neighboring pixels in a circular set with pre-defined radius	Text localization in HDIs by building three basic reference models (background, text and image). By analyzing the extracted textural features and thereafter by measuring model responses, blocks were labeled and classified. The textural features were extracted from a partition of a DI into logical grids with fixed size disjoint window.	-Easiness of adaptation to document complexity by adding many more reference models -High values were obtained to localize text regions for the classification accuracy metrics: 88.60% and 93.61% of precision by using the LBP and variance features, respectively	-Not parameter-free -Block re-labeling with hierarchical multi-resolution analysis using pre-defined rules
[235]	Zipf law	-Gray-scale graphics and drop caps (printed) -French and Latin	Similar drop caps responses to a request drop cap	-Size of the mask used as the pixel neighborhood -k classes of the k-means quantization task -Number of slopes in the Zipf plot -Distance used in the indexing step (the Hamming distance in the parameter space)	Indexing drop caps by analyzing the Zipf law. The Zipf law builds a model to characterize the distribution of patterns occurring in the drop caps. Then, three meaningful values associated to the Zipf plot were automatically extracted representing three splitting points in the Zipf curve segment. Finally, each drop cap image was represented by the three extracted features.	Simple and efficient	-Need to restrict the number of perceived patterns and to have a relevant model by considering a smaller pixel neighborhood (*i.e.* 4-connectivity) and using a clustering algorithm of the gray-levels in k classes -Requirement to apply a histogram normalization filter on the images to ensure better use of the image spectrum
[236]	-Fractal dimensions -Points of interest	-Entire gray-scale or color pages (printed) digitized at 300 dpi -Arabic and Latin	Retrieved image which corresponds to the image request	Number of clusters	Categorization and matching of HDIs based on the fractal dimensions and points of interest by using the scale-invariant feature transform (SIFT).	-Experiments were carried out on 1000 images -Fast approach due to the use of points of interest	-Not sufficiently efficient on high degraded HDIs -Need to apply a denoising step by using a Gaussian filter which can lead to a loss of relevant information -Use of a suitable filter to remove noise in HDIs which depends on the degradation/noise level
4- Spectral methods							
[1]	GFs	-Entire gray-scale or color book page (printed) digitized at 300 dpi -French and Latin	Pixels of book content labeled as graphics, text or background	-Size of the analysis windows -Number of pre-defined Gabor frequencies and orientations -Number of clusters	Detection of the text whatever its orientation.	-Without formulating an assumption on document layout or content -CLARA was applied as a sampling-based clustering method which has the advantage to deal with large datasets -No need to create training data and train a model (*i.e.* pixel features were clustered into homogeneous regions) -Optimal joint localization properties of the Gabor features in both the spatial and frequency domains	-High cost of processing time and memory resources -Need to pre-define the number of clusters for the clustering step and to assign label to each resulting cluster -Application of the clustering task on all pixel book pages (*i.e.* foreground and background pixels) -Dependency of the clustering performance on the selected samples used on the CLARA clustering algorithm (*i.e.* trade-off of efficiency) -Pre-defined Gabor parameters (orientations and spatial frequencies)

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
[237]	GFs	-Collected binary, gray-scale or color character images and pages of word set (printed) -Portuguese	Classified characters	-Gabor wavelet direction -Wavelet orientation -Frequency of GF banks	OCR based on oriented features extracted using GFs (12 wavelet directions were considered, from 0° to 180°). By considering sets of character images and combining their dominant oriented graphical features which were extracted from the GF banks, the fuzzy membership functions were generated. Then, the classification of new character images can be processed. Besides the Gabor features, the image aspect ratios for the characters were also extracted and analyzed.	-Algorithm suited for handwritten recognition in HDIs -88% of mean character recognition rate -Use of the fuzzy classification improves the recognition rate results by providing larger tolerance -Large test set (*i.e.* 8034 characters) was used in the experiments which consisted of 20 pages acquired with variable scanning conditions (e.g. skewing and paper see-through, with both non-italic and italic text) -Optimal joint localization properties of the Gabor features in both the spatial and frequency domains	-Need for a training step with collected character image samples -Pre-set Gabor parameters (orientations and spatial frequencies)
[238]	GFs	-Entire binary or gray-scale pages (printed periodicals) -Arabic	Nets	-Number of Gabor orientations -Size of the structural element for erosion -Minimal net length -Filter or selection threshold	Extraction of different types of nets (e.g. slightly erased lines or lines with inclinations and curvatures) from binarized DIs. Two GFs were applied to the binary image, one with the orientation 0° to detect horizontal nets and another one with 90° to segment vertical nets. Many post-processing steps were introduced after exploring GFs (e.g. erosion CC analysis).	-High performance to detect nets tainted with white areas, nets overlapping with text (nets that touch with text areas) and nets with a high inclination degree or with curvature -25% and 22% of gain in the impurity rate (under-segmentation) and incompleteness rate (over-segmentation), respectively, compared to the existing approach -Less sensitive to nets quality degradations and noise (discontinuous, with burrs or partially erased nets) -Optimal joint localization properties of the Gabor features in both the spatial and frequency domains	-Use of the Niblack adaptive binarization algorithm -Based on *a priori* knowledge -Parametrization of GFs (orientations and spatial frequencies) -Use of several post-processing steps (e.g. morphological operators as a classical filtering technique and CC analysis technique) -Inability to identify certain types of real nets (e.g. strongly erased or too short nets) and some classes of counterfeit nets (tears, linear marks due to the digitization) -High execution time
[239]	GFs	-Image patches -Arabic, Chinese and Cyrillic	Identified text blocks	-Spatial frequencies and direction of GFs -Size of the image patches -Dimensionality of the tensor subspace	Text block identification from HDIs based on the image patches analysis (IPA) and the Gabor feature extraction. Firstly, DIs were partitioned into small patches without overlapping. Subsequently, positive and negative patches were selected to compose an active training set.	-Use of MDA guarantees a significant gain in computation time, memory and performance -Selection of useful features provides satisfactory identification results -Capture of local texture features of each patch and the global information of the training data due to the use of IPA	-Pre-defined Gabor parameters (orientations and spatial frequencies) -Requirement for a supervised learning phase -Need to resize the images and to select region of interest to reduce noise pixels on the edges of the image

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
				-Number of iterations for the multi-linear discriminant analysis (MDA) -Number of trees for the random forest classifier	Then, the Gabor features were extracted on each patch to characterize the analyzed text blocks. The MDA was applied to reduce the dimensionality of the data. Finally, a random forest classifier was learned on the selected Gabor features after selecting automatically the informative features.	-Optimal joint localization properties of the Gabor features in both the spatial and frequency domains -Use of the random forest classifier was simple, easily paralleled and relatively robust to outliers and noise (*i.e.* it gives useful internal estimates of error, strength, correlation and variable importance) -Good evaluation on Chinese, Arabic and Cyrillic document scripts	
[240]	GFs	-Entire color ancient manuscripts -Arabic	Detected main text area	-Spatial frequencies and direction of GFs -Connectivity of the refinement step -Constant defined on the distance computation of the refinement step -Coarse binary mask in GF computation to approximate the rectangular shape of the main text area	A learning-free approach to detect the main text area from side-notes in ancient manuscripts based on coarse-to-fine scheme. First, a coarse segmentation of the main text area was processed by using GFs. Then, the segmentation was refined by formulating the problem as an energy minimization task and achieving the minimum using graph cuts.	-Promising results in terms of segmentation quality (*i.e.* 98.84% of mean F-measure was noted on 38 HDIs) and time performance (*i.e.* $01' 13''$ per page on average) -Learning-free approach and does not include a local refinement step	-Pre-defined Gabor parameters (orientations and spatial frequencies) -Experiments were only conducted on a set of 38 HDIs -Requirement for refinement segmentation step -Use of a coarse binary mask in GF computation to approximate the rectangular shape of the main text area -Use of a training phase
[241]	-Multi-scale shape decomposition -Curvelet Transform	-Text regions (gray-scale or color manuscripts) -French	Image with a similar writing style to the original request image	No parameter	Writer classification based on curvelet features in relation to the two discriminative shapes properties (curvature and orientation). Curvature and orientation descriptors were extracted from the curvelets to generate a compact signature for each writing. Then, a similarity measure was defined to compare two handwriting samples by retrieving images from the database that have a similar writing style to the original request.	-Adaptability to different manuscripts corpus characterized with different writing styles and shapes properties -78% and 89% of precision on the whole Middle-Ages and humanistic database, respectively	More difficult to separate Medieval handwriting styles

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
[242]	-Wavelet transform -Multi-scale analysis	-Entire gray-scale or color pages (printed and manuscripts) -Arabic, Latin and Hebrew	Blocks of historical pages clustered as graphics, text or background	-Number of wavelet decomposition levels -Size of the analyzed block -Size of the analysis windows -Number of clusters	Text/graphics/background separation to characterize images of HDIs for a possible physical segmentation and different types of alphabet segmentation (Arabic, Latin and Hebrew) were processed by extracting the wavelet features (with three decomposition levels) from each selected page block. A multi-scale analysis technique was applied to each analyzed block by selecting concentric and non-overlapping windows for the wavelet feature extraction. Through the reliefF algorithm and the factor analysis technique, irrelevant and redundant features were eliminated. Finally, the k-means algorithm was applied on the selected wavelet features to cluster page blocks.	Selection of relevant wavelet features to reduce the storage space and to increase the clustering performance	-Experiments were carried out on limited corpus resources (*i.e.* twenty HDIs) -No quantitative evaluation (*i.e.* they only gave visual results) -User intervention is necessary to set the number of clusters when using the k-means algorithm for page block clustering
[243]	SP transform	-Entire complex multi-lingual multi-script pages (degraded binary, gray-scale or color official administrative documents) -French and Arabic	Regions classified into text (machine-printed or handwritten) and non-text (images, graphics, drawings or paintings)	-Smearing parameter settings for document mask generation -Number of clusters -Number of scales and directions of the SPs	Segmentation of complex multi-lingual multi-script documents: separation text/graphics and extraction of graphics, tables and text lines. The SP features were extracted to locate and classify regions into text (machine-printed or handwritten) and non-text (images, graphics, drawings or paintings) from noise-infected, deformed, multi-lingual and multi-script DIs. The textural features were extracted from pre-segmented regions which were obtained by applying the morphological operators (e.g. merging characters to obtain text regions).	-Handling multi-script documents -Invariant to skew, rotation and translation -Working fairly well on different kinds of documents -Adapted to complex layout characterized by different kinds of clutters	-Use of the Otsu's global thresholding for document binarization -Use of the morphological operators for document denoising -Failed to locate text blocks that are not well separated from the background or are connected to graphics -Classification failed when some words or text blocks have size and shape similar to graphics ones (e.g. titles with large font size) -Not parameter-free -Need for spatial-domain implementation
[244]	1-level wavelet transform using 3-tap Daubechies filter	-Entire color pages (manuscripts) -Arabic	Blocks classified into background, text or graphics	-Number of clusters -Size of the analysis block	Foreground/background separation and text/graphics segmentation. By applying the multi-scale analysis technique using the 1-level/3-tap Daubechies wavelet transform on the luma information of blocks of 32×32 pixels, an image for the background and another one for the foreground were produced.	Encouraging results for background/foreground separation	-Not parameter-free -Segmentation in several successive stages which led to segmentation errors in each step

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
				-Number of neighboring pixels -Size of the analysis window for relaxation phase	Then, by extracting statistical features extracted from the 1-level/3-tap Daubechies wavelet transform of the luma information of foreground blocks of 32×32 pixels, analyzing the extracted features using the fuzzy C-means algorithm and introducing a relaxation step as a post-processing step, the discrimination of the foreground layers of the document, particularly of two classes: text and graphics was achieved.		-Introduction of a relaxation/post-processing step for refinement of the background/foreground segmentation results -Unsatisfactory results of segmenting graphics due to the high complexity of DIs
5- Hybrid methods							
[1]	-Frequency attributes -Auto-correlation function -Rose of directions -Multi-scale analysis	-Entire gray-scale or color book page (printed) digitized at 300 dpi -French and Latin	Pixels of book content labeled as graphics, text or background	-Size of the analysis windows -Number of clusters	For separation of page elements using textural descriptors, three auto-correlation features were extracted which were derived from the rose of directions and two frequency attributes by using a multi-scale analysis for classifying HDI pixels into text, graphics and background. Then, by using the CLARA clustering algorithm, the extracted texture descriptors were clustered and pixels were separated into different content clusters.	-83% and 92% were noted as mean good classification rates for the graphics and text pixels, respectively -Without formulating an assumption on document layout or content -CLARA was applied as a sampling-based clustering method which has the advantage to deal with large datasets -No need to create training data and train a model (*i.e.* pixel features were clustered into homogeneous regions)	-High cost of processing time and memory resources -Need to pre-define the number of clusters for the clustering step and to assign label to each resulting cluster -Application of the clustering task on all pixel book pages (*i.e.* foreground and background pixels) -Dependency of the clustering performance on the selected samples used on the CLARA clustering algorithm (*i.e.* trade-off of efficiency)
[4]	-Gabor dominant orientation histogram (GDOH) -Rotation invariant uniform LBP operators	-Entire gray-scale or color pages (handwritten historical manuscripts) -Latin, German and English	Pixels of HDIs labeled as periphery, background, text block or decoration	-Number of Gabor orientations -Number of neighboring pixels in a circular set with pre-defined radius -Size of the analysis window -Scale factor of the analyzed image at each level in the pyramidal approach -Number of levels in the pyramidal approach	A physical structure detection method for historical handwritten DIs was proposed by classifying and labeling each pixel as periphery, background, text block or decoration. Without any assumption of specific topologies and shapes, coordinates, color and texture features (e.g. color variance, smoothness, Laplacian, LBP, GDOH) were used for classification. Then, the FCBF algorithm was applied to select relevant features and remove redundant ones and to reduce the feature size without degrading the classification accuracy. Finally, the segmentation results were refined by a smoothing post-processing task.	-Without formulating any assumption of specific topologies and shapes -High performance of HDI segmentation by combining coordinates, color and texture features -Effective and robust to changes of writing style, page layout and noise on HDIs	-Not parameter-free -Segmentation at several levels in the used pyramidal approach which led to segmentation errors -Use of feature selection and post-processing steps

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
				-Number of clusters at each level in the pyramidal approach -Fixed threshold to measure the predominant correlation for the FCBF algorithm -Pre-defined threshold value which corresponds to the number of text line pixels in the neighborhood of the analyzed pixel in the post-processing task			
[29]	-Co-occurrence -Run-length matrices -Auto-correlation function -Wold decomposition	Gray-scale drop caps (printed)	-Drop caps retrieved according to an input of a query drop cap -French and Latin	-Range of the GLCM distance and direction values -Size of the sliding window for the global segmentation -Ratio of the analyzed small blocks by the entire image into small blocks for the local segmentation -Number of clusters -Distance used (the Bhattacharyya distance) in the drop cap retrieval step	By extracting several texture image features (GLCM, run-length matrices, auto-correlation function and Wold decomposition) on a sliding window at segmented areas of interest of drop caps, signatures were computed for drop cap indexing. Firstly, a global segmentation was applied by using the GLCM and the average of its uniformity to partition of a drop cap into homogeneous regions and texture regions were generated after applying a binary threshold. Secondly, a local segmentation was performed to investigate the coarseness and fineness of texture by optimizing the theoretical auto-correlation function to the modeled one derived from the Wold decomposition and extracting textural features from the GLCM and the run-length matrices.	-Wold decomposition ensures a good interpretation of a texture as a combination of multiple signals and description of the fineness and coarseness of texture -Experiments were conducted in 344 gray-scale drop cap images	-Need to apply a binary threshold to the extracted GLCM uniformity for segmenting drop caps into homogeneous and texture regions -Numbers of clusters are known in advance -Requirement for a morphological operation to reduce noise and eliminate small regions from the segmented regions of each layer -No quantitative evaluation of the segmentation rate due to the lack of the ground-truth of drop caps -No criteria to measure the performance of the proposed drop cap retrieval system

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
					Afterwards, the k-means clustering algorithm was performed to provide final segmentation by assuming the numbers of clusters were known in advance. Then, the MST and pairwise geometric attributes (PGA) were applied for the drop cap signature generation by considering the three segmented layers (*i.e.* homogeneous, texture and contour layers). Finally, for drop cap retrieval from the database by using a query drop cap, the Bhattacharyya distance was used.		
[30]	-Hermite transform -Gabor transform	-Binary, gray-scale or color text regions (manuscripts) -French	A list of images that were ordered according to their similarity with the query which was the income of characterized and classified handwritings	-Length of the Krawtchouk filters -Parameters of the Hermite filters (the maximum derivatives of order D (or polynomial degree) and scale) -Translation value -Size of the analysis window -Number of selected salient handwriting directions extracted from the rose of directions and the Gabor scales	Writing enhancement, background noise, text/drawing separation and handwritten patterns characterization with orientation features in ancient manuscripts using the Hermite and GFs. For noise reduction, the Hermite filters analyzed the image in the frequency domain. After noise reduction, the image was reconstructed and GFs were parametrized to detect relevant handwriting orientations using the analysis of the rose of directions. A signature for each analyzed handwritten image was produced based on the computation of the Gabor features. Finally, a similarity measure was defined to compare different samples.	-No need for segmenting text in characters, graphemes or to localize it precisely (*i.e.* segmentation-free) -Analysis of both global feature and local shape properties (texture properties of handwriting and local oriented variations along pattern contours) -91% of correct classification with the correct class as first response -Two handwriting extracts of different sizes (and also of different writing sizes) can be compared -Optimal joint localization properties of the Gabor features in both the spatial and frequency domains	-Assumptions had to be made to distinguish handwritings on the front side from the noise on the background (threshold value used on the Hermite Filters depended on the original contrast) -Size of the localization window for noise reduction step can affect the selection of the frequency range used for the Hermite filter analysis -Use of orientation features will not be relevant if the background is textured too much and if it contains too many oriented noisy strokes or when the samples contain very badly written texts with too many irregularities (*i.e.* non-constancy of a same writer) -Need for a normalized text density with nor empty areas neither noisy strokes line regions in the analyzed handwriting blocks (*i.e.* text contains quantitatively significant handwritten patterns that are estimated by two extreme entropy values) -Dependency of performance on the quality of the noise reduction step -Pre-defined Gabor parameters (orientations and spatial frequencies)

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
[64]	-GFs -CRF -RLF (relative location features)	-Entire color pages (handwritten historical manuscripts (the 5CofM2 database)) -Old Catalan	Document blocks classified into three classes	-Number of pre-defined Gabor frequencies and orientations -Number of clusters	Document segmentation method based on the RLF to segment structured HDIs collected from the 5CofM2 dataset into three classes: the family name, the record body and the paid tax. The RLF technique included in the CRF framework and combined with Gabor features, was used to segment text regions into three classes.	Good results for segmenting text regions and detecting of each of the three classes	-Not parameter-free -Pixel-level approach (*i.e.*, high execution time) -Applicable to structured DIs -Requirement for a training task and a testing step -Fixed thresholds to configure the CRF on the training datasets -Need for graph cut algorithm to perform energy minimization of the CRF
[57]	-GFs -Rotation invariant uniform LBP operators	-Entire gray-scale or color pages (handwritten historical manuscripts) -Latin, German and English	Text lines	-Number of Gabor orientations -Number of neighboring pixels in a circular set with pre-defined radius -Size of the analysis window -Scale factor of the analyzed image at each level in the pyramidal approach -Number of levels in the pyramidal approach -Number of clusters at each level in the pyramidal approach -Fixed threshold to measure the predominant correlation for the fast correlation-based filter (FCBF) algorithm -Pre-defined threshold value which corresponds to the number of text line pixels in the neighborhood of the analyzed pixel in the post-processing task	A text line segmentation algorithm applicable to color historical manuscripts was proposed based on topographical, color and texture features. GFs and rotation invariant uniform LBP operators were used in a pyramidal approach to classify pixels into: text, background, decoration and out of page. For text line segmentation, a modified FCBF algorithm was proposed to remove automatically redundant and irrelevant features before applying a support vector machine (SVM) classifier. Finally, after text line segmentation a post-processing task was applied to refine the results of labeling pixels into two classes: text line and non-text line.	-Modular algorithm allowing the evaluation of different features and classifiers -No need for script-specific knowledge -Applicable to color HDIs -Robust and adapted to different writing styles, page layouts, *etc.*	-Not parameter-free -Segmentation at several levels in the used pyramidal approach which led to segmentation errors -Requirement for a training task and a testing step -Use of feature selection and post-processing steps -Need for a pre-processing task to remove noise on the borders of text lines (*i.e.* sensitive to noise) and another post-processing step to validate pixels on the borders of text lines

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
[84]	-Compactness -Elliptic degree -Angular signature -Generic Fourier descriptors -R-signature -Fourier–Mellin -Moments of Zernike	-Drop caps (printed) -French and Latin	Recognized drop caps	-Number of clusters -Pre-set threshold for angular signature extraction	By assigning to the extracted descriptors a recognition map and defining automatically a descriptor measure for each cluster of samples, the characterization of a learning set can be built. Then, by combining several descriptors, the recognition rate of drop caps extracted from archival documents was improved.	-Fast feature extraction task -Improvement of the recognition rate by combining different features	-Requirement for a classical algorithm of binarization, based on an entropy criterion calculated on the gray-level histogram for the extraction of object and background clusters -Need to apply the dilation and erosion steps to clean the region -Requirement for a set of rules and measures (size, compactness, *etc.*) to extract the drop cap
[172]	-Black/white transitions -Entropy -Compactness -Tiling -Histogram entropy -Eccentricity -Number of CCs, *etc.*	-Segmented and pre-labeled text lines from binary or gray-scale DIs (archives of Savoie and scientific journals) -French	Classified layout elements (e.g. capital letters, italic text)	-SVM kernel parameters -Tiling surface -Pixel density limit	Characterization and classification of segmented and pre-labeled text lines from DIs by extracting textural features and investigating discriminant power using SVM classifiers.	100% recognition rate of homogeneous blocks written in bold, italic, capital letters, *etc.*	-Requirement for physical segmentation at a line level as a pre-processing step -Need for a number of support vectors for the learning phase which is equal to the number of input patterns
[211]	-Histogram entropy -Statistical moments	-Entire color typed pages (Nabuco's bequest) digitized at 200 dpi -Latin	Regenerated documents with similar texture to the original documents	-Two multiplicative factors for entropy rule definition -Number of lines -Number of neighbor pixels -Multiplicative value for texture generation rules	Generating paper texture of HDIs by applying firstly an entropy-based algorithm to segment the image of document into the image of the paper background and the printing of the document. Then, to generate the texture of the paper, statistical moments were computed to fill in the gaps from the printing, yielding a blank sheet of paper with similar texture to the original document.	Satisfactory results of document regeneration from the texture produced with added ink	-Empirical definition of two multiplicative factors for entropy ranges and rules -Experiments were only conducted on a set of 50 samples from Nabuco's bequest -Pre-defined number of neighbor pixels and multiplicative value for texture generation rules which were experimentally determined
[245]	-Circular statistics description of a directional histogram -Color histograms (red, green and blue color space (RGB), and enhanced hue, saturation and value space (HSV))	-Entire color high resolution digitized images (manuscripts) -Italian	Classified pixels, blocks and regions	-Number of text lines to analyze the texture within the square block -Size of the analysis window -Rate of the analysis window overlap	Automatic manuscript layout segmentation and extraction of valuable pictures from the decorated pages by combining several categories of texture features. The directional histogram feature was computed using a polar representation of the auto-correlation matrix for text segmentation.	-Automatic extraction of valuable pictures from the decorated pages by means of visual cues, independently by the layout -Use of the GSDM reduces the training requirements of learning algorithms both in terms of the number of samples and the computational time, without impacting on the classification performance	-Use of a learning phase -Extraction of an important number of features for each block which led a high-dimensionality of the feature space (*i.e.* 1028-dimensional feature vector) -Requirement for few post-processing steps or relaxation labeling tasks (e.g. filling isolated blocks to force a neighborhood consistency or removing smallest blobs when using the CC analysis technique)

Table 3.2 – continued from previous page

Ref.	Tool	Input/Language	Output	Parameter	Description	Advantage	Disadvantage
	-Gradient spatial dependency matrix (GSDM)			-Range of the distance and direction values of GSDM	Then, a proposed textural feature, namely GSDM, aimed at detecting the correlations between the gradient directions by means of visual cues, was computed for image areas extraction. RGB histogram and enhanced HSV histogram as color features or visual descriptors were extracted from the pictorial regions of the page to distinguish the semantic content of the different decorative parts. Finally, a clustering-based embedding process was afterward used to reduce the training requirements of learning algorithms, and to classify and separate feature vectors which were extracted for each pictorial block (*i.e.* pictures and decorations).	-Classification of blocks produced a precision of 85.8% by using a clustering-based embedding approach and by combining the RGB histogram, enhanced HSV histogram and GSDM descriptors	
[246]	-Run-lengths (horizontal and vertical) -CRF	-Entire pages (degraded newspapers archives) -French	Pixels classified into different functional entities (e.g. titles and sub-titles, graphical separators, text lines, columns	-Quantization feature functions -Optimization algorithms -Number of neighbors in the horizontal window	Structure extraction from old newspapers by defining an horizontal CRF model dedicated to pixel labeling. Each pixel was characterized by its horizontal and vertical run-lengths. Then, the contextual features between labels have been introduced into the CRF model as template to take into account the horizontal dependencies of the label and each computed run-length and to give afterward quantized feature functions.	Good results for text line extraction task and particularly for curved/degraded text lines.	-Use of a learning phase -Use of multiple quantization feature functions as a pre-processing step to provide discrete observations extracted from the whole image to the CRF -Use of multiple optimization algorithms when the CRF model is trained -Dependency of the number of features on the observation set size and the number of feature templates

Table 3.3.: Texture-based methods reviewed for document layout analysis by Okun and Pietikäinen [6].

Ref.	Tool	Input	Output	Parameter	Description	Advantage	Disadvantage
[173]	TCS	Binary or gray-scale image	Rectangular blocks	None	Co-occurrence of pixel values were extracted within a window centered at each pixel and analyzed by the nearest neighbor to classify regions into text, image, *etc.*	-Parameter-free -Adapted to both binary and gray-scale images	Specific to extract rectangular blocks
[174]	Laws masks	Gray-scale image	Labeled bitmap	Number of the stationary hidden Markov models (HMM)	By using HMM based texture analysis, one model for each texture type was produced and trained for text/textured-background separation. From block-based to pixel-based segmentation, the extracted textural features were accurately analyzed.	High performance in text extraction from complex textured background	Computationally expensive in memory and processing time
[189]	GFs	Gray-scale image digitized at 75 dpi	Bounding boxes placed around detected rectangular regions for text blocks or labeled bitmap	-Spatial frequencies -Orientations -Number of clusters, *etc.*	Gabor features were extracted and clustered to separate text from halftone pictures. By detecting the CCs and finding the bounding boxes of the rectangular regions, text pixels were clustered into larger regions.	-Skew insensitive -Optimal joint localization properties of the Gabor features in both the spatial and frequency domains	-High computationally expensive in memory and processing time -Pre-defined Gabor parameters (orientations and spatial frequencies)
[247]	White tiles	Binary image	Array of classified white tiles	Four pre-defined parameters related to white tile computation	Segmentation and classification of an image using white tiles and texture features. The texture features were extracted from the segmented white tiles (e.g. number and area of white tiles).	-Robust for complex shaped regions -Skew insensitive	Deal only with binary images
[248]	Texture masks	Gray-scale image digitized at 100 dpi	Bounding boxes placed around detected regions	-Number of pixel samples -Two global thresholds when applying a binarization task	Selection of a set of texture masks that minimized the classification error when segmenting halftones, background and line-drawing regions was applied by using a neural network approach in the training step. Clustering and post-processing tasks were applied on the extracted texture features for separating text from line-drawing regions.	Discriminant power of the extracted attributes to separate text of different languages based on the connectivity analysis technique	Slow due to the important size of the training set
[249]	Structured wavelet packet analysis	Gray-scale image digitized at 200 or 300 dpi	Labeled set of windows	-Number of window pixels -Shift pixel between the adjacent windows	Low-order moments of wavelet packet components were used as texture features by adopting a multi-scale technique with a soft classification approach.	Robust to unconstrained document layout and page skew	Requirement for extra tasks to obtain larger regions from the classified windows

Chapter 4.

A texture feature benchmarking for historical document image analysis

This chapter presents an experimental evaluation and benchmarking of a number of commonly and widely used texture features which have been conducted on a large corpus of historical document images.

Contents

4.1. Introduction

It is commonly agreed that texture analysis plays a fundamental role for HDI analysis and understanding since it has been considered as a consistent choice for meeting the need to segment a page layout under significant degradations and different noise levels and types. In addition, it has been shown that texture-based approaches work effectively with no *a priori* knowledge about the layout, content, typography, font styles, scanning resolution, image size of the document *etc.* It has also been proved that they have good performance even for skewed images and handwritten text. However, faced with a large diversity of the texture-based methods, few questions arise. Which texture methods are firstly well suited for segmenting graphical regions from textual ones, discriminating text in a variety of situations of different fonts and scales and separating different types of graphics ? Then, which texture approaches represent a constructive compromise between the performance (*i.e.* segmentation quality) and computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality) ? It is well-known that the success or failure of texture-based segmentation method tightly depends on the type of the extracted and used texture features. Thus, an experimental evaluation and benchmarking of a number of commonly and widely used texture approaches have been firstly conducted on a large corpus of HDIs to have satisfactory and clear answers to the above questions. Thus, in this chapter, an experimental evaluation of a number of commonly and widely used texture features has been conducted on a large corpus of HDIs for the purpose of determining the performance of each texture-based feature set according to the document content, *i.e.* segmenting graphical regions from textual ones on the one hand and discriminating text in a variety of situations of different fonts and scales on the other hand. To provide a qualitative measure of which texture-based feature sets are most appropriate for this task, nine texture-based feature sets (Tamura, local binary patterns (LBP), gray-level run-length matrix (GLRLM), auto-correlation function, gray-level co-occurrence matrix (GLCM), Gabor filters (GFs), 3-level Haar wavelet transform, 3-level wavelet transform using 3-tap Daubechies filter and 3-level wavelet transform using 4-tap Daubechies filter) have been investigated and assessed on 1100 pages of historical documents by using a classical texture-based pixel labeling scheme for comparing the texture features. The results reported in this chapter provide a useful benchmark in terms of performance, texture vector dimensionality, memory requirements, processing time and complexity for current and future research efforts in HDI analysis.

The remainder of this chapter is organized as follows: Section 4.2 reviews the different surveys and comparisons of texture-based techniques proposed in the literature, with a particular focus on those related to DIA and HDIA. Section 4.3 presents a brief description of the different texture-based feature sets evaluated in this work. In Sections 4.4 and 4.5, we outline the experimental protocol by describing the experimental corpus, the defined ground-truth and the used pixel labeling scheme for comparing the texture features. In addition, we discuss the obtained performance of texture feature analysis experiments by computing several clustering and classification metrics for an evaluation of accuracy. Qualitative results are also given to demonstrate the performance of each texture-based feature set, along with the computational cost (*i.e.* resources in terms of the memory requirements, complexity and time consumption considerations and texture vector dimensionality). Our discussion and conclusions are presented in Sections 4.6 and 4.7, respectively.

4.2. A short review of surveys and comparisons of texture-based techniques

Numerous surveys and comparisons of texture-based techniques have been proposed for image segmentation and analysis in the literature a few years ago. For example, Weszka *et al.* [250] compared different texture analysis methods based on the Fourier power spectrum, second-order gray-level statistics and first-order statistics of gray-level differences for terrain classification. They concluded that the first and second order statistics perform significantly better than the spectral

approaches. A well-researched survey and complete overview of recent texture segmentation and feature extraction techniques for unsupervised applications was presented in [142], including GFs, GLCM, fractals, *etc.* They concluded that texture-based methods have distinct applications, *i.e.* some model-based texture methods are suitable for stochastic textures, while some spectral-based texture methods (e.g. GFs) are adequate for stochastic and structural textures. However, they did not present a quantitative comparison of the surveyed texture-based methods since they stated that is a demanding and time-consuming task. Few limited studies attempted to present quantitative comparisons of texture-based algorithms [250, 158, 251, 252]. Myint *et al.* [253] compared the effectiveness of the wavelets, fractals, auto-correlation and GLCM for urban mapping using high spatial resolution remote-sensing images. They concluded that the auto-correlation and GLCM approaches are relatively effective when compared to the fractal ones. They also proved that the wavelet transform approach is the most accurate of the four investigated approaches. Chang *et al.* [254] compared three different sets of texture features: GLCM, Law's texture energy and Gabor multi-channel filtering for segmentation of homogeneous regions of real scene images. Subsequently, they compared three clustering techniques for segmentation of homogeneous regions (fuzzy c-means clustering (FCM), minimum square-error clustering (CLST) and split-and-merge) based on the computed texture feature values. They concluded that the choice of a clustering technique influences the texture segmentation results. Moreover, they proved that the Gabor approach with the CLST technique has the best performance.

However, there are limited comparative studies of texture-based methods in the most explored DIA fields. For instance, Busch *et al.* [200] evaluated a number of commonly used textures features, including the GLCM, Gabor energy and a number of wavelet features by extracting energy, logarithmic mean deviation, logarithmic co-occurrence and scale co-occurrence for determining the script of a DI. Experimental results showed that the logarithmic co-occurrence features give the lowest overall classification error rate, while the GLCM descriptors give the highest overall classification error rate. A few comparative studies of Gabor and co-occurrence features for script and language identification [199] and DI segmentation [173] have been proposed. More comparisons can be found concerning Gabor and gradient features for character recognition [197, 198]. Nourbakhsh *et al.* [2] evaluated two texture-based approaches (GFs and log-polar wavelets) for separating text/non-text in DIs. Baâti *et al.* [255] compared three texture-based approaches (GFs, GLCM and wavelets) for Arabic/Latin and printed/handwritten script differentiation. They concluded that the GLCM outperformed the Gabor and wavelets approaches. He *et al.* [201] evaluated three approaches based on GFs, discrete wavelet transform and contourlet for handwriting-based writer identification. For Arabic font recognition, GFs, GLCM, wavelet transform using 2-tap Daubechies filter (Db2) wavelet and SP transform were compared in [256]. An outperformance of SP transform was obtained with a high recognition rate approximately equal to 99%.

Okun and Pietikäinen [6] presented a survey of seven texture-based methods to review the progress achieved for DI layout analysis. The seven analyzed texture-based methods are based on the following texture features: run-lengths [169], multi-channel GFs [189], TCS [173], white tiles [247], texture masks [248], structured wavelet packet analysis [249] and laws masks [174]. The reviewed methods were evaluated on magazines and newspapers (gray-scale or binary images). The majority of texture-based methods used within this survey assumed that the image backgrounds of the analyzed DIs are white with the exception of that used by Chen [174] which aimed to separate text from textured background. A summary table of these reviewed texture-based methods, describing briefly their algorithms, parameters, inputs and outputs, and showing their pros and cons, are presented in Table 3.3.

In spite of invaluable number of different texture-based studies and contributions has been achieved on different sub-fields and tasks of pattern recognition, there is a very limited number of comparative studies of texture-based approaches in the fields of DIA and particularly historical DIA. Those texture-based approaches have been reported as relevant and dedicated to a specific application and fine-tuned to a particular dataset. Thus, the interest to texture-based algorithms

is increasing continuously for historical DIA. Indeed, during the last two decades, several texture-based feature sets have been investigated and demonstrated robust when they have been extracted and analyzed from degraded and unconstrained DIs [228]. It has also been proved that these methods work effectively with no *a priori* knowledge [1]. Nevertheless, the question of how these texture-based algorithms are compared with each other has not been properly addressed for historical DIA. This is mostly due to the unavailability or lack of a standard public dataset of HDIs and its associated ground-truth [39].

Faced with such diversity of texture-based methods, few questions arise. Which texture features are firstly well suited for segmenting graphical regions from textual ones, discriminating text in a variety of situations of different fonts and scales and separating different types of graphics ? Then, which texture features represent a constructive compromise between the performance (*i.e.* segmentation quality) and the computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality) ? It is well-known that the success or failure of texture-based segmentation method tightly depends on the type of the extracted and used texture features. Our choice of the different texture-based feature sets to investigate and compare (Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor and wavelets), basically statistical, frequency and model-based methods, is justified by the following reasons: Firstly, we have made a comparative study about selecting the texture feature category which ensures the best trade-off between the best performance, the reduced number of parameter settings and thresholds and the lowest computation time (*cf.* Table 3.2). Secondly, the extraction of these texture features needs less parameter settings. Indeed, without hypothesis on either the DI layout or content, the choice of numerous appropriate thresholds and parameters is a very difficult task. Then, the texture descriptors such as the Tamura [214], LBP [234], GLRLM [215], auto-correlation [1], GLCM [175], Gabor [257] and wavelet [249] features have been widely investigated for a long time in independent experiments in order to extract texture features and segment and characterize DIs or part of them. In addition, they have been proved relevant and robust to noise, unconstrained DI layout, page skew, *etc.* The Gabor and wavelet-based approaches have been known to be relevant and are widely used for many fields of DIA even they seem high resource-consuming ones. Nevertheless, the GLCM and GLRLM approaches are identified as the best choices when the numerical complexity is taken into account. Moreover, the LBP-based approach has been known to be a model-based approach which is characterized by a low computational complexity which has been used recently for segmentation of historical machine printed DIs [234]. Besides, the Tamura features which have been known to be a classical ones, they have the advantage to guarantee that the space generated from them is perceptual uniform. Finally, the high performance of segmenting HDIs based on the auto-correlation function [245, 1, 230, 229, 89], leads us to investigate and analyze the auto-correlation features.

4.3. Texture features

The texture-based feature sets which have been assessed in this work are extracted from the Tamura, LBP, GLRLM, auto-correlation, GLCM, GFs and three wavelet-based (Haar, Db3 and Db4) approaches. The following provides a brief description of the different extracted texture features. Nevertheless, in Appendix B and particularly in Section B.1, an exhaustive and detailed review of the different analyzed texture features has been carried. First, for each set of texture descriptors a state-of-the-art related to the parametrization of the used texture features in the most explored fields in image analysis and pattern recognition, with a particular focus on those related to subfields and tasks of DIA and historical DIA, is briefly presented. Then, a detailed review of the texture features and their parameters is discussed. Finally, we conclude by detailing and justifying the techniques and parameters used in our study based on work published in the literature and after performing several experiments to choose the best configuration of the pre-defined thresholds and parameters. The different analyzed texture features in this work are summarized in Table 4.1.

4.3.1. Tamura

The first set of texture features investigated in this work is the Tamura descriptors. Tamura *et al.* [159] proposed to extract textural features corresponding to human visual perception. They defined six basic texture descriptors, namely coarseness, contrast, directionality, line-likeness, regularity and roughness. They proved that the three first textural features (*i.e.* coarseness, contrast and directionality) consistently outperformed others for global descriptions of textures both separately and in combinations for image segmentation and classification issues.

Recently, the Tamura descriptors have been extracted to assist DIA. Keysers *et al.* [214] compared several texture features, including the Tamura texture features histogram, relational invariant feature histogram, run-length histogram, distribution of connected components, *etc.* for DI zone classification. They concluded that the Tamura features are the single best ones but they have high demand in computational time (*i.e.* more than 100 times slower to compute than the most other extracted descriptors). Mouats *et al.* [258, 259] introduced the Tamura descriptors into their Gabor-based segmentation of HDIs method to improve the obtained results.

Four Tamura descriptors are extracted in this work, namely:

- Coarseness (*cf.* equation B.4),
- Contrast (*cf.* equation B.5),
- Number of orientations (*cf.* equation B.11),
- Directionality (*cf.* equation B.12).

In Appendix B and particularly in Section B.1.1, a detailed description of the different extracted Tamura features has been carried.

4.3.2. LBP

The second set of texture features investigated in this work is the LBP descriptors. The LBP operator is one of the most explored local image descriptor for texture analysis which has mainly used for describing local texture properties of gray-scale images. It has been introduced to measure pure and original property of the texture spectrum by Wang and He [260]. They proposed a texture analysis pattern based on a texture unit. LBP is a two-level version of the texture spectrum method. Later, it was popularized by Ojala *et al.* [261] and Harwood *et al.* [262] to analyze texture characteristics for texture classification. Ojala and Pietikäinen [263] presented an unsupervised texture segmentation method based on examining the LBP distributions.

LBP is obtained by locally thresholding texture and their combinations with local gray-scale measures. It represents each analyzed image pixel with a binary pattern based on the difference between its gray-level value and its circular neighborhood with specified radius R_l. If the gray-level value difference between the analyzed pixel $I_c(x, y)$ and its P_l neighboring pixels $I_{p\in[0,P_l-1]}(x, y)$, is greater than or equal to zero, the LBP value is set to 1, otherwise is set to 0. In this work, a rotation invariant uniform 2 LBP operator which is labeled $LBP^{riu2}_{P_l,R_l}$, is used. For describing an image with $LBP^{riu2}_{P_l,R_l}$, a histogram of binary patterns $Hist_{P_l,R_l}$ of P_l+2 bins is produced. Each bin provides an estimation of the probability to find the corresponding pattern in the analyzed image.

Recently, the LBP operator has gained great attention of many researchers in the DIA fields. Dua *et al.* [264] extracted the LBP wavelet domain for off-line and text-independent writer identification. Lutf *et al.* [265] proposed a LBP-based approach for writer identification using off-line Arabic handwriting. They computed the LBP histogram to extract handwriting features for each diacritic after retrieving all diacritics from the input image. Ferrer *et al.* [266] proposed an algorithm based on the LBP orientation for printed script identification. Since Nicolaou *et al.* [204, 205] worked on binary images as inputs, they presented an approach based on appropriate redundant oriented binary LBP operator for Arabic font recognition. Bhowmik and Kar [234] compared the

rotation invariant uniform LBP operator with the variance measure for segmentation of historical machine printed DIs. They concluded that the LBP operator outperforms the variance measure for separating graphic regions from text ones.

Jiang *et al.* [206] used the $LBP_{P_l=8,R_l=1}$ operator for printer identification. They generated 59-dimensional histogram (a feature vector composed of 58 uniform patterns and 1 single non-uniform pattern) from the LBP operator for each analyzed gray-scale pixel of a DI. Bertolini *et al.* [203] extracted the LBP features from the $LBP^{u2}_{P_l=8,R_l=2}$ operator for writer identification and verification. They proved that the used LBP operator which produces a feature vector of 59 components for each analyzed pixel, is fast and accurate. Nicolaou *et al.* [204, 205] introduced a redundant oriented LBP ($P_l = 8, R_l = 3$) for Arabic font recognition. They extracted 327 redundant LBP features, including 255 bins from the LBP histogram, 36 rotation invariant features, 8 rotation phase features, 14 edge features, 5 beta-function features and 9 sample-count features. Bhowmik and Kar [234] localized text in HDIs by extracting LBP_{P_l,R_l}, $LBP^{ri}_{P_l,R_l}$ and $LBP^{riu2}_{P_l,R_l}$ features. They used three LBP operators by setting R_l equal to 1, 2 and 3 and P_l equal to 8, 16 and 24, respectively. But, they considered only P_l equal to 8 during the binary pattern computation. They concluded that the LBP_{P_l,R_l} model outperforms slightly the two other models $LBP^{ri}_{P_l,R_l}$ and $LBP^{riu2}_{P_l,R_l}$. But, in the most cases, the obtained results of the three models are relatively similar.

In this work, we set P_l and R_l equal to 8 and 1, respectively. Thus, for each image pixel $I_c(x, y)$, $LBP^{riu2}_{8,R_l}(I_c(x, y))$ produces 10 $Hist_{P_l,R_l}$. The number of the uniform and non-uniform patterns are 9 and 28, respectively, to ensure better discrimination of spatial patterns. Indeed, 10 $LBP^{riu2}_{P_l=8,R_l=1}$ descriptors are extracted. The $LBP^{riu2}_{P_l=8,R_l=1}$ feature vector consists of 10 terms of the probability to find the corresponding pattern in the analyzed image. The nine first descriptors correspond to the nine $Hist_{P_l=8,R_l=1}$ bins which represent the uniform patterns (*cf.* equation B.18), while the last one represent the last $Hist_{P_l=8,R_l=1}$ bin which characterizes all the non-uniform patterns (*cf.* equation B.19). Therefore, the $LBP^{riu2}_{P_l,R_l}$ features are:

- Heights of the uniform bins of the $Hist_{P_l=8,R_l=1}$ (*cf.* equation B.18),
- Height of the non-uniform bin of the $Hist_{P_l=8,R_l=1}$ (*cf.* equation B.19).

In Appendix B and particularly in Section B.1.2, a detailed review of the LBP operator and LBP features has been carried.

4.3.3. GLRLM

The third set of texture features investigated in this work is the GLRLM descriptors. The GLRLM descriptors are extracted by applying the run-length method. The run-length method has been extensively studied in a wide array of fields for analysis of images and particularly for pattern recognition and texture classification [267]. It has been introduced by Galloway *et al.* [181] to classify a set of terrain samples by extracting various run-length features from several GLRLM.

For a given image, an element of the GLRLM $p(g, l)$ is defined as the number of runs with pixels of gray-level g and run-length l. A gray-level run g is a sequence in a scan direction of a set of consecutive and collinear image pixels with identical gray-level value. The length of the run l is the number of image pixels in the run. A GLRLM is computed for runs having any given direction. Usually, the four scan directions have been used: $\theta_r = \{0°, 45°, 90°, 135°\}$ (*i.e.* horizontal, vertical, diagonal and anti-diagonal directions). For the GLRLM, the dimension of g is equal to G^l which corresponds to the maximum gray-level (*i.e.* number of gray-level bins). On the other hand, the dimension of l is equal to L which corresponds to the maximum run-length. Afterwards, a 2-D run-length histogram ($Hist_{g,l}$) is produced for each scan direction, such one axis represented the run-length and the other axis illustrates the gray-level value or gray-level value bin. $Hist_{g,l}$ is a histogram of run-lengths.

Although the poor performance of using the run-length or GLRLM features obtained by Weska *et al.* [250], and Conners and Harlow [268] compared to classical texture features (GLCM, gray-level

difference and the power spectrum features), the run-length methods have been recently applied to meet the need for DI segmentation or DIA, *etc.* Seuret *et al.* [223] proposed a method for discriminating printed content from handwritten annotations at pixel level. They extracted the run-length features in four directions $\theta_r = \{0°, 45°, 90°, 135°\}$ to estimate the width of a stroke in a given direction. Stamatopoulos *et al.* [269] used the run-length method for the page frame detection from double page DIs. They detected the vertical and horizontal zones of the two pages based on the vertical and horizontal white run projections, respectively. Nikolaou *et al.* [127] proposed an adaptive RLSA for the text line, word and character segmentation of historical and degraded machine-printed DIs. Although the proposed algorithm has been proved to work efficiently for a wide variety of degraded DIs, several thresholds were defined in the used segmentation techniques. Shi and Govindaraju [134] used a fuzzy run-length approach for the line separation in complex handwritten DIs including postal parcel images and historical handwritten DIs. Keysers *et al.* [214] proposed an accurate system for the classification of DIs based on the run-length feature extraction. The extracted features were used to classify text/non-text DI zones. Gordo *et al.* [215] used the multi-scale binarizing run-length histograms for the large-scale DI retrieval and classification. They worked on binary images as inputs, they quantized the lengths of the runs in logarithmic scale by defining 9 intervals for each quantized level (*i.e.* black and white gray-levels). Then, four run-length histograms were computed in horizontal, vertical, diagonal and anti-diagonal directions for each extracted region using spatial pyramids. The four run-length histograms were concatenated to characterize the extracted regions by a region descriptor of length $72 = 4$ directions $\times$ 2 quantized levels $\times$ 9 quantized intervals. The extracted descriptors have been proved that they work efficiently and do not require *a priori* knowledge of the DI layout, model, content or any kind of layout analysis. Dinstein and Shapira [270] extracted textural features based on the run-length histograms from groups of characters for the ancient Hebraic handwriting identification. The horizontal and vertical directions were selected to compute the run-length histograms. Then, the average dissimilarity between histograms of each writer was defined. Experiments yielded satisfying results. Another algorithm based on the run-length features was proposed for the handwriting identification on medieval DIs [271]. Uttama *et al.* [29] examined drop caps from historical heritage images and introduced a drop cap segmentation method based on a combination of different texture features extracted from the GLCM, GLRLM, auto-correlation function and Wold decomposition. Three run-length descriptors were extracted, including long-run emphasis (LRE), run percentage (RPC) and gray-level distribution.

In this work, for each analyzed foreground pixel, four 2-D run-length histograms ($Hist_{g,l}$) are produced for each scan direction $\theta_r = \{0°, 45°, 90°, 135°\}$, *i.e.* horizontal, vertical, diagonal and anti-diagonal directions. For each 2-D run-length histograms $Hist_{g,l}$, a feature vector of 11 terms of GLRLM indices is computed. The 11 texture features based on gray-level run-lengths and particularly the 2-D run-length histogram ($Hist_{g,l}$) are introduced by Galloway *et al.* [181] to capture the coarseness of a texture in a specific direction:

- Short-run emphasis (SRE) (*cf.* equation B.21),
- Long-run emphasis (LRE) (*cf.* equation B.22),
- Low gray-level emphasis (LGRE) (*cf.* equation B.23),
- High gray-level emphasis (HGRE) (*cf.* equation B.24),
- Gray-level non-uniformity (GLNU) (*cf.* equation B.25),
- Run-length non-uniformity (RLNU) (*cf.* equation B.26),
- Run percentage (RPC) (*cf.* equation B.27),
- Short-run low gray-level emphasis (SRLGE) (*cf.* equation B.28),

- Long-run high gray-level emphasis (LRHGE) (*cf.* equation B.29),
- Short-run high gray-level emphasis (SRHGE) (*cf.* equation B.30),
- Long-run low gray-level emphasis (LRLGE) (*cf.* equation B.31).

In Appendix B and particularly in Section B.1.3, a detailed description of the different extracted GLRLM features has been carried.

4.3.4. Auto-correlation

The fourth set of texture features investigated in this work is the auto-correlation descriptors. The auto-correlation features are extracted from a non-parametric tool which consists of the auto-correlation function. The auto-correlation function which is a 2-D function, is defined as a similarity measure between a dataset and a shifted copy of the data. It is used to find periodic patterns and similar patterns through a number of extracted auto-correlation features [145, 179].

The auto-correlation descriptors highlight interesting information on the principal orientations and periodicities of texture allowing characterizing the content of DIs without any assumption on the page layout, content, DI typographical or graphical characteristics. The use of the auto-correlation function is not new for the DIA community. Numerous studies have identified a number of auto-correlation features for segmenting HDIs and contemporary DIs [30, 1, 230, 245, 272, 229, 89]. Eglin *et al.* [30] determined the number of bank of GFs by selecting the relevant directions which were deduced from the rose of directions, to select interesting patterns for the noise reduction and classification of handwritings in ancient manuscripts. For historical DIA, Journet *et al.* [1] defined three auto-correlation features which few ones were derived from the rose of directions. The extracted features computed over the neighborhood of each pixel (foreground and background), were as follows: the main orientation of the rose of directions, the intensity value of the auto-correlation function for the main orientation and the variance in the intensities of the rose of directions, except for the main orientation. Grana *et al.* [245] used the auto-correlation matrix to distinguish between textual and pictorial regions in historical manuscripts. Garz and Sablatnig [230] presented a multi-scale texture-based approach for text region recognition in ancient manuscripts. They extracted the three auto-correlation features proposed firstly by Journet *et al.* [1] by applying three scales by means of overlapping sliding windows. Ouji *et al.* [272] introduced two other texture attributes (*i.e.* mean stroke width and height of an image), also in relation to the auto-correlation function for contemporary DI segmentation. For geometric layout analysis of HDIs, Coppi *et al.* [229] extracted the main regions from the page using the RXYC algorithm, then each region was divided in small squared blocks, and the local auto-correlation features were computed on each block and classified using a SVM classifier. The local auto-correlation features were deduced from a directional histogram obtained from the projections of the auto-correlation matrix along the vertical and horizontal directions in order to identify the repeating pattern of the texture. A 308-dimensional feature vector for each block was constructed.

Five auto-correlation features are extracted in this work [1, 272]:

- Main orientation of the rose of directions (*cf.* equation B.35),
- Intensity of the auto-correlation function for the main orientation (*cf.* equation B.36),
- Variance of the intensities of the rose of directions (*cf.* equation B.37),
- Mean stroke width along specific directions (*cf.* Algorithm 8),
- Mean stroke height along specific directions (*cf.* Algorithm 9).

where the rose of directions which is a derivative of the auto-correlation function, is deduced from the auto-correlation function [273].

In Appendix B and particularly in Section B.1.4, a detailed description of the different extracted auto-correlation features has been carried.

4.3.5. GLCM

The fifth set of texture features investigated in this work is the GLCM or co-occurrence attributes [180]. The GLCM or co-occurrence matrix is a classic of statistical texture-based segmentation methods. The GLCM is an estimate of the second order probability density function of image pixels. This matrix determines the probability of occurrence of pixel pairs according to their gray-levels and distance by considering the spatial relationship of pixels in the image.

A GLCM element is the probability of the gray-level pairs defined in a specified direction θ_c and separated by a particular distance of d_c units. The co-occurrence descriptors are then statistics computed from the GLCM. They provide second order statistical information of neighboring pixels of an image. Multi-distance and multi-direction can be applied to extract a large number of GLCM descriptors. Usually, the co-occurrence matrices are generated for a small range of distance values $d_c = \{1, 2\}$ and typically for the directions $\theta_c = \{0°, 45°, 90°, 135°\}$ [200].

A number of other works based on the GLCM feature extraction and analysis have also been proposed in order to segment and classify the content of DIs [274, 275]. More methods based on the GLCM feature analysis have been proposed in the literature for identifying script and language from DIs [276, 200]. For Arabic font recognition, the GLCM with $d_c = 4$ for 4 orientations $\theta_c = \{0°, 45°, 90°, 135°\}$ were used in [256]. Usually, the co-occurrence matrices are generated for a small range of distance values $d_c = \{1, 2\}$ and typically for the directions $\theta_c = \{0°, 45°, 90°, 135°\}$ [200]. A survey of DI segmentation methods using texture analysis presented different methods for segmenting DIs [173]. A texture analysis approach based on the assembly of n^{th} order co-occurrence information within a processing window was also proposed. This study stated that the GLCM approach is the best one in terms of processing time and complexity. For segmenting DI contents into text, graph, table and picture, Kim and Kim [175] analyzed six standard GLCM features (entropy, contrast, energy, uniformity, diagonal moment and homogeneity) in the entropy image.

In this work, from the computed co-occurrence matrices, eight GLCM features are extracted for two distances $d_c = \{1, 2\}$ [274, 200]:

- Maximum entry in the GLCM or maximum probability (*cf.* equation B.43),
- Correlation metric (*cf.* equation B.44),
- Energy or angular second moment (*cf.* equation B.45),
- Entropy (*cf.* equation B.46),
- Inertia or contrast (*cf.* equation B.47),
- Local homogeneity (*cf.* equation B.48),
- Cluster shade (*cf.* equation B.49),
- Cluster prominence (*cf.* equation B.50).

In addition to the 16 co-occurrence features (eight for each distance), two other descriptors are computed (mean value (*cf.* equation B.51) and standard deviation (*cf.* equation B.52) of the energy) for the two combined distances [275]. The 18 extracted GLCM features have been shown to perform well for script identification in [200].

In Appendix B and particularly in Section B.1.5, a detailed description of the different extracted GLCM features has been carried.

4.3.6. Gabor

The sixth set of texture features investigated in this work is the Gabor descriptors. The Gabor features are extracted using the multi-channel Gabor filtering technique. The original Gabor elementary functions have been firstly proposed by Gabor [277]. The multi-channel Gabor filtering is inspired by the multi-channel filtering theory which has been first investigated by Campbell and Robson [278] for the visual information processing of the human visual system. Daugman [279] modeled the visual information processing of the human visual system by the 2-D multi-channel Gabor functions which are local spatial band-pass filters. The main idea of the multi-channel filtering technique is to exploit the differences in dominant sizes and orientations of different textures by decomposing the original image into several filtered images with limited spectral information. The 2-D Gabor functions have the advantage to have the conjoint resolution information in both the 2-D spatial and Fourier domains. The filtered images are proceeded by tuning the analyzed image to combinations of frequency and orientation in a narrow range which are referred to channels and interpreted as band-pass filters. By applying a bank of GFs, the specified channels cover the spatial-frequency domain.

A 2-D GF is a linear selective band-pass filter, dependent on two parameters (spatial frequency f_g and orientation θ_g) which characterize the specified channel. It consists of a Gaussian kernel function modulated by a sinusoidal plane wave. The spatial frequency f determines the distance from the Gaussian centers to the origin while the orientation θ_g specifies the angle from the horizontal axis (*i.e.* α-axis to the Gaussian centers). The multi-channel Gabor filtering approach is inherently multi-resolutional which are close relatives of the wavelet transform [218].

Texture features generated by GFs have been increasingly considered and applied to DIA. During the last two decades, Gabor-based analysis approaches have been proposed for biometric identification based on handwriting [280, 156, 281], writer identification [282], handwritten word recognition [283], character recognition [284], font recognition [285], script identification [286, 287], signature recognition [288], palm print recognition [289], degraded DI binarization [290], *etc.* Zhu *et al.* [285] proposed a texture-analysis-based algorithm for automatic font recognition by extracting the Gabor features. They noted a $99,1\%$ of mean recognition rate. Ma and Doermann [257] proposed a GF-based multi-class classifier in order to identify scripts, and font faces and styles. A binarization method based on Gabor filter bank for ancient degraded DIs was proposed in [290]. A GF bank with four orientations (0, $\pi/4$, $\pi/2$ and $3\pi/4$) weighted by the dominant foreground script slant angle of the DI and one selected frequency was used to determine more efficiently the foreground information.

Nevertheless, numerous approaches have been sought for text segmentation and extraction from digital DIs using the Gabor descriptors [189, 291, 292, 191]. Several studies have been conducted in the literature for page layout analysis using the multi-channel GFs [293, 257, 294], while few ones have explored GFs for HDI segmentation. For instance, Ribeiro *et al.* [237] proposed an optical character recognition (OCR) system for HDI analysis and recognition by applying fuzzy methods on aligned oriented features extracted using GFs in the training step. Vieux and Domenger [216] proposed a pixel-based classification approach to separate text from other classes (e.g. illustrations and background) by using a bank of GFs at five scales (1, $\sqrt{2}$, 2, $2\sqrt{2}$ and 4) and six orientations ($k\frac{\pi}{6}, k \in \{0,\ldots,5\}$). Their approach was evaluated on a public dataset containing magazines and technical journals. They found 86%, 82.7% and 53.7% of F-measure for segmenting background, text and graphic pixels, respectively. Jain *et al.* [248] showed the effectiveness of applying a multi-channel Gabor filtering-based texture segmentation approach for segmentation and classification of DIs. They chose the five following spatial frequencies: $4\sqrt{2}$, $8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$ and $64\sqrt{2}$. Charrada and Ben Amara [238] extracted nets from ancient Arab periodicals by exploring GFs. Zhong and Cheriet [239] used the dimensionally reduced multi-channel GFs for text block identification on image patches from HDIs. They extracted 28 GFs from image patches in their experiments, where 7 spatial frequencies ($\sqrt{2}$, $2\sqrt{2}$, $4\sqrt{2}$, $8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$ and $64\sqrt{2}$) and 4 orientation angles (0, $\pi/4$, $\pi/2$ and $3\pi/4$) were pre-defined. Cruz-Fernández and Ramos-Terrades [64] computed a 36-

dimensional Gabor feature vector for each analyzed pixel using 9 orientations (0, $2\pi/9$, $4\pi/9$, $6\pi/9$, $8\pi/9$, $10\pi/9$, $12\pi/9$, $14\pi/9$ and $16\pi/9$) and 4 spatial frequencies (an overlapping degree of 0.5 in the frequency domain with the highest frequency is equal to 0.35) for structured HDI segmentation. For Arabic font recognition, 16 Gabor channels were computed with 4 frequencies $f_g = \{8, 16, 32, 64\}$ and 4 orientations $\theta_g = \{0°, 45°, 90°, 135°\}$ in [256]. A learning-free approach to detect the main text area from side-notes in ancient manuscripts based on coarse-to-fine scheme [240]. A coarse segmentation of the main text area was processed by using GFs. The proposed approach achieved promising results in terms of segmentation quality (*i.e.* 98.84% of mean F-measure was noted on 38 HDIs) and time performance (*i.e.* $01' 13''$ per page on average). The four directions (0, $\pi/4$, $\pi/2$ and $3\pi/4$) are widely used in the literature [189, 248, 285, 257].

In this work, the magnitude response of the output of Gabor functions is investigated. The magnitude of the output is important if the specified GF matched the particular texture, otherwise low response to the specified GF corresponds to poor match of the dominant texture properties of the analyzed image to the set of the spatial-frequency components of the fixed GF [295]. 24 GFs are applied (6 different spatial frequencies f_g=$\{2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}$ and $64\sqrt{2}\}$ and 4 different orientations θ_g=$\{0, \pi/4, \pi/2$ and $3\pi/4\}$) (*cf.* Figure B.18). The space of GF is set constant $\sigma_g = \sigma_x = \sigma_y = 1$. When convolving an image with 24 Gabor channels (obtained by using 6 different spatial frequencies and 4 different orientations), 24 Gabor filtered images are produced. In this work, 24 responses of filtered images or Gabor responses are generated. Finally, by convoluting the analyzed whole DI at each specified channel defined by a pair of orientation and frequency, the Gabor features are extracted from the magnitudes of the Gabor filtered images. The extracted Gabor features represent the statistical distribution of the Gabor magnitude response. They consist of two simple statistics:

- Mean value of the Gabor filtered magnitude response corresponding to all pixels defined in the analyzed sliding window of the filtered image (*cf.* equation B.54),
- Standard deviation of the Gabor filtered magnitude response corresponding to all pixels defined in the analyzed sliding window of the filtered image (*cf.* equation B.55).

In Appendix B and particularly in Section B.1.6, a detailed description of the different extracted Gabor features has been carried.

4.3.7. Wavelet

The last set of textural features examined in this work is the wavelet descriptors. Mallat [154] investigated the application of the wavelets as multi-resolution representations to data compression in image coding, texture discrimination and fractal analysis. The wavelet features which are extracted from the wavelet transform provide interesting insight on the statistical characteristics of the analyzed image. The wavelet features represent consistent properties in the localization of the frequency space and multi-resolution.

A 2-D wavelet transform ensures the localization in both the scale (frequency) domain via dilations and in the time domain via translations of the mother wavelet. A 2-D wavelet transform represents an image with both the spatial and frequency characteristics. The objective of a 2-D wavelet transform is to decompose an image into low and high frequency sub-band images (*i.e.* to filter out several frequencies range). The 2-D J-level wavelet transform decomposes a discrete input image $I(x, y)$ into 4 sub-bands and it produces $3J + 1$ sub-images $(A_{2^{-J}}, \{D^{(v)}_{2^{-j}}, D^{(h)}_{2^{-j}}, D^{(d)}_{2^{-j}}\}_{j=1,2,\ldots,J})$. J represents the scale of the discrete wavelet transform. j denotes the decomposition level of the discrete wavelet transform such as $j = 1, 2, \ldots, J$. $A_{2^{-J}}$ is the approximation of the input image $I(x, y)$ at 2^{-J} resolution. $D^{(v)}_{2^{-j}}$, $D^{(h)}_{2^{-j}}$ and $D^{(d)}_{2^{-j}}$ are 3 detail components of the input image $I(x, y)$ at 2^{-j} resolution. The wavelet coefficients in $D^{(v)}_{2^{-j}}$, $D^{(h)}_{2^{-j}}$ and $D^{(d)}_{2^{-j}}$ illustrate the vertical, horizontal and diagonal high frequencies, respectively.

Recently, a lot of studies of applying the wavelet transform have been reported for many fields of DIA. The wavelet transform has been very effective for DI pre-processing [296], watermarking [208], handwriting-based writer identification [201], script identification [200, 255], text localization [217, 297], page segmentation [212], printer discrimination [207], *etc.* Maatouk *et al.* [208] showed that the 3-level decomposition with the Db2 and Db3 family provided the best performance for the watermarking of HDIs. Kricha *et al.* [296] proposed a denoising step by applying a thresholding technique in the coefficients of wavelet sub-bands to reduce the noise in the background of HDIs. Furukawa [207] used the bi-orthogonal spline 2 wavelet transform for discriminating printers based on contours qualities of printed characters. For script recognition, Busch *et al.* [200] evaluated a number of wavelet features based on energy, logarithmic mean deviation, logarithmic co-occurrence and scale co-occurrence. Baâti *et al.* [255] used the energy of 12-level bi-orthogonal wavelet coefficients for script identification. Hiremath and Shivashankar [298] also extracted features from the co-occurrence histograms of wavelet decomposed images for script identification. They concluded that the Haar wavelet yields the best classification results. Manthalkar *et al.* [299] also computed the rotation and scale invariant texture features using the discrete wavelet packet transform for script identification. They evaluated two wavelet families (bi-orthogonal and Daubechies) and they concluded that the bi-orthogonal wavelet outperforms the Daubechies ones (*i.e.* 83.07% and 80.89% of overall correct classification for the bi-orthogonal and Daubechies wavelets, respectively). Pardeshi *et al.* [222] extracted the directional multi-resolution information based on the Daubechies9 wavelet transform to automatically identify automatic handwritten Indian scripts. For the handwriting-based writer identification, He *et al.* [201] used the 3-level wavelet transform using a 4-tap Daubechies filter. Many studies applied the 3-level wavelet transform by using a 3-tap Daubechies filter to identify Arabic font [300, 301, 302, 303]. Gazzah and Ben Amara [304] explored the 2-D discrete wavelet transform based on a lifting scheme for writer identification (off-line Arabic handwriting). They compared 9 wavelet families, including the three following Daubechies wavelets (Daubechies2, Daubechies3 and Daubechies5), 4 Cohen-Daubechies-Feauveau wavelets, lazy wavelet transform and Symlet wavelets. They reported that the different evaluated wavelets give similar results (equal to 95%). He *et al.* [305] compared GFs with a wavelet approach based on the generalized Gaussian density for the off-line handwriting-based writer identification. They showed that the proposed approach based on the wavelet transform performs better than the traditional 2-D GFs and it is better in terms of processing time. Ding *et al.* [306] used the 3-level spline2 wavelet transform on the normalized image of a single Chinese character for the character independent font recognition. Zhang *et al.* [307] performed a statistical analysis on the stroke patterns obtained from the wavelet decomposed sub-images using a 2-tap Symlet filter for the italic font recognition. For Arabic font recognition, the wavelet energy (*i.e.* sum of square of the detailed wavelet transform coefficients) was extracted from the Daubechies2 wavelet transform in [256]. Angadi and Kodabagi [308] extracted texture features (the zone wise wavelet energy features, vertical run statistical features of the wavelet coefficients and wavelet logarithmic mean deviation) from the wavelet transform for the word level script identification of text in the low resolution display board images. For multi-font Arabic character analysis and the extraction and classification of the handwritten shapes from ancient manuscripts, derivative forms of the wavelet transforms (e.g. ridgelet, curvelet and contourlet transforms) have been used [309, 241]. These specific wavelets offer the best trade-off between local and global features for handwritten recognition.

For page segmentation, Gupta *et al.* [212] studied the energy distribution over different scales of the orthonormal wavelet decomposition. Li and Gray [219] investigated the distribution characteristics of the wavelet coefficients of the 1-level Haar transform for DI segmentation. They noted that the results produced by the two longer wavelet filters (4-tap Daubechies and 8-tap Daubechies) are similar while the Haar transform has the best localization property since its filter is the shortest and it has the least processing time. They extracted two features related to the pattern distribution of the wavelet coefficients using the Haar wavelet transform instead of computing moments of the wavelet coefficients as features. The first descriptor defines the rate of fit goodness of the

distribution of the wavelet coefficients in high frequency bands to the Laplacien distribution. Then, the second feature determines the concentration rate of the wavelet coefficients in high frequency bands at few discrete values. They noted a 4.1% of average classification error rate. Kumar *et al.* [310, 217] compared the Haar discrete wavelet transform and matched wavelet for text extraction and DI segmentation. Liang and Chen [297] suggested to use the Haar discrete wavelet transform for the text region extraction from the static images or video sequences. They showed an average error rate close to 1.42%. Acharyya and Kundu [311] presented a multi-scale analysis method based on the wavelet scale-space features using a 8-tap filter for the text segmentation in DIs. Nourbakhsh *et al.* [2] used the log-polar wavelet energy signatures for the text localization and extraction from the complex gray-scale DIs. Jin and Tang [312] proposed an approach to determine the positions of the text areas in the complex-background images using the wavelet decomposition. Etemad *et al.* [249] presented an algorithm based on the pyramidal wavelet transform and wavelet packet tree using the Daubechies filters for the segmentation of unstructured DIs. A wavelet-based technique has been proposed for the reference line extraction from gray-level background DIs in [313]. For the text/non-text segmentation in DIs, Deivalakshmi *et al.* [314] extracted the wavelet-based GLCM features. The evaluated wavelet transforms are: Haar, Db4, Db25, Symlet8, Coilflet3 and Coilflet5. The Coilflet5 wavelet transform used in their algorithm outperforms the five other investigated wavelets. An average classification rate equal to 92.97% has been obtained with using the Coilflet5 filter. Kricha and Ben Amara [242] explored the correlation between the different sub-bands of the same decomposition level and the auto-correlation of each sub-band in the wavelet transform for the text/graphic separation in HDIs and the discrimination of the different alphabet kinds (Arabic, Latin and Hebrew). They computed the 1-order and 2-order statistics performed from the correlation function of each analysis window. Subsequently, they took into consideration only the mean and standard deviation of the auto-correlation of the approximation sub-band obtained from the 3-level decomposition of the wavelet transform and performed at four different sizes of analysis windows in order to adopt a multi-scale approach.

The Haar and Daubechies wavelets are the most used ones since they have been proved to work effectively in many applications. The Haar wavelet transform is the fastest among all wavelets since its coefficients are either 1 or -1. Thus, they are the less complex, simplest and most widely used wavelets, while the Daubechies ones are characterized by the fractal structures [297, 315].

Therefore, in this work the wavelet features are extracted from the 2-D 3-level discrete stationary wavelet transform with a limited number of taps (3-level wavelet transform using Haar filter (Haar), 3-level wavelet transform using 3-tap Daubechies filter (Db3) and 3-level wavelet transform using 4-tap Daubechies filter (Db4)) (*cf.* Figure B.21). Therefore, 10 sub-bands ($A_{2^{-3}}$, $D^{(v)}_{2^{-1}}$, $D^{(h)}_{2^{-1}}$, $D^{(d)}_{2^{-1}}$, $D^{(v)}_{2^{-2}}$, $D^{(h)}_{2^{-2}}$, $D^{(d)}_{2^{-2}}$, $D^{(v)}_{2^{-3}}$, $D^{(h)}_{2^{-3}}$ and $D^{(d)}_{2^{-3}}$) are generated.

In our experiments, in order to reduce the number of the wavelet coefficients, two simple statistics deduced from the wavelet transform coefficients for each sub-band are extracted to form feature vector of 20 terms (10 sub-bands). They represent the statistical distribution of the wavelet coefficients. The two simple statistics:

- Mean value of the wavelet transform coefficients for each sub-band defined in the analyzed sliding window of the image (*cf.* equation B.60),
- Standard deviation of the wavelet transform coefficients for each sub-band defined in the analyzed sliding window of the image (*cf.* equation B.61).

In Appendix B and particularly in Section B.1.7, a detailed review of the wavelet features has been carried.

4.4. Experimental protocol

We have experimented the nine texture-based feature sets on a wide variety of HDIs and on different HDI content types for assessing the discriminating power of the extracted features. In this section,

a brief description of the main phases of a proposed classical pixel-labeling scheme for comparing texture features is presented. Subsequently, the performance of each texture-based feature set is detailed after describing our experimental corpus and its associated ground-truth and presenting the used accuracy metrics for the performance evaluation.

4.4.1. Pixel-labeling scheme for comparing texture features

The texture feature extraction has the objective to reduce information in DI content to a set of descriptive textural features. The extraction of textural descriptors helps to describe the DI layout and content by analyzing the texture feature space computed from the extracted textural characteristics of DI content (*i.e.* by mapping the differences in the spatial structures of each digitized DI into differences in gray-level value for each page). However, different results are shown according to the specified extracted kind of texture used for segmenting or characterizing the DI layout on the one hand, and the DI content on the other hand. Therefore, in this work our goal is determining the performance of each texture feature set according to the DI content and providing an additional insight into the computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality) of each analyzed texture feature set. However, there is a real need for a generic and standard framework that permits a fair comparison of texture features. For this purpose, a standard pixel-labeling scheme for comparing texture features is proposed in this work (*cf.* Figure 4.1). This scheme is considered as the support of this comparative study or benchmarking of the nine different texture-based feature sets.

Since our objective is to find regions with similar textural content from DIs characterized by a wide variety of contents, layouts and shapes, we opt for a pixel-based approach due to its advantage to overcome the limits and constraints of region and boundary-based approaches (*cf.* Section 3.3.2) [3]. Baird *et al.* [316] reported that analyzing the DI content by classifying individual pixels, not regions, has the advantage to get away from the dependence on the arbitrariness and restrictiveness of the limited families of region shapes. They proposed a pixel-based classification approach based on investigating 26 textural features, all extracted from the luminosity channel (e.g. region luminosity average, line luminosity average, line average difference, line luminosity average difference, line luminosity max difference, revised distance to max-difference pair, revised distance to max-difference pixel). The DI pixels were classified into machine-printed text, handwritten text, photographs or blank space. They reported a low per-pixel accuracy equal to 62.4%. Seuret *et al.* [223] proposed a method for discriminating the printed content from handwritten annotations at pixel level. They extracted from the foreground pixels and their neighbors several features (mean luminosity, luminosity variance, smoothness, gradient density, arithmetic operators, shannon's entropy, histogram moments, edge detectors, GLCM, side histogram and run-length). The foreground and background pixels are separated with the Sauvola's binarization algorithm [317]. A method for selecting the optimal window size for each feature was afterwards introduced. Then, the multilayer perceptron (MLP) technique was applied on the computed features to classify the foreground pixels. Finally, a post-processing step was introduced to corrects the typical mis-classifications by removing outliers based on several heuristics. A 96.10% of mean accuracy was noted. Vieux and Domenger [216] proposed an hierarchical clustering model to learn and classify pixels in DIs (magazines and technical journals). They extracted the Gabor features to separate text from illustrations or other pre-defined classes. They concluded that by using a pixel-based approach, the performance is independent of the accuracy of pre-processing steps such as the binarization or segmentation. For DI segmentation, Vil'kin *et al.* [318] extracted 26 texture features (e.g. mean brightness feature, several textural descriptors computed from the GLCM) from various positions (*i.e.* tiles, a small block inside a large one and small overlapping blocks) and different sizes of DI blocks. They compared subsequently four supervised classification algorithms. They noted 85%, 86% and 86% of rates of correctly classified pixels for the three variants of arrangement of blocks (*i.e.* tiles, a small block inside a large one and small overlapping blocks), respectively. Nevertheless, they pointed out that the use of small overlapping blocks led to more accurate segmentation at the

cost of the processing time. Journet *et al.* [1] proposed a pixel-based method by using a multi-scale analysis (*i.e.* textural descriptors were extracted from pixels of the gray-level DIs at four different sizes: (32×32), (64×64), (128×128) and (256×256)) for the pixel-clustering of HDI content into text, graphics or background. Each pixel was characterized by five textural features computed at four different scales (20 indices).

Kise [5] proposed a general processing flow which has been described by the following four tasks:

1. Pre-processing step such as noise reduction,
2. Texture feature extraction from each pixel of an input gray-scale or color image,
3. Classification of generated textural feature vectors,
4. Post-processing stage.

In order to analyze and evaluate the different texture-based feature sets, a generic, standard or classical pixel-labeling scheme for comparing texture features is proposed in our experiments (*cf.* Figure 4.1).

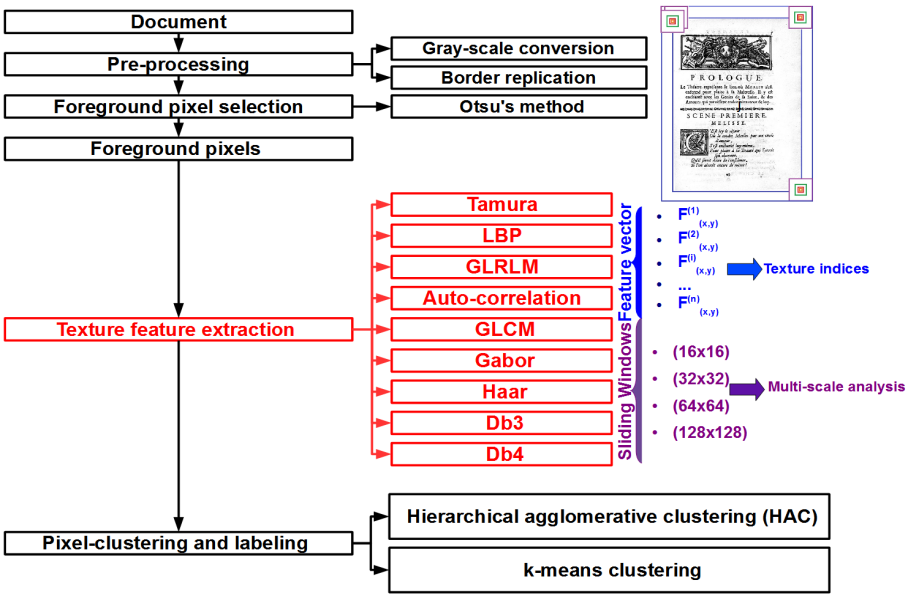


Figure 4.1.: Pixel-labeling scheme for comparing texture features.

The pixel-labeling scheme for comparing texture features is conceptualized by three modular processes:

1. ***Pre-processing and foreground pixel selection*** (*cf.* Section 4.4.1.1),
2. ***Texture feature extraction*** (*cf.* Section 4.4.1.2),
3. ***Pixel-clustering and labeling*** (*cf.* Section 4.4.1.3).

In this work, we are not looking for an accurate segmentation, but to find regions with similar textural content as easily, quickly and automatically as possible. It has been largely proved that the proposed texture tools are relevant for DIA and characterization. But, it can neither segment a DI into graphics, paragraphs, *etc.* nor characterize its structure (e.g. columns, rows, paragraphs). The region segmentation and classification tasks can be carried at the end after introducing a post-processing phase by taking into consideration the topological or spatial relationships (e.g. hierarchy, inclusion, neighborhood position). The proposed pixel-labeling scheme for comparing texture features has the possibility to be extended for consequent DI processing such as region segmentation

and classification, by introducing a standard post-processing method (e.g. morphological cleaning approach, multi-scale majority voting technique). Nevertheless, in this study our goal is to find the best texture feature sets for discriminating the textual regions from graphical ones and separating different text fonts without taking into account the spatial relationships between pixels, *i.e.* without introducing a post-processing stage [319].

In addition, due to a possible bias produced by performing a classification task, this step is not included in this work by applying a training phase through supervised machine learning tools (*i.e.* by using a set of training pixels with corresponding known labels, a pixel classification model can be applied). Therefore, the pixel classification and post-processing tasks are beyond the scope of this comparative study. Nevertheless, if we produce a relevant pixel-labeled DI, homogeneous regions will be identified by the page segmentation stage and will be labeled according to the content type by the region classification stage. Indeed, the pixel-labeling task is necessary for further data processing by different techniques since it provides the basis for all subsequent segmentation, analysis, classification and recognition processes such the OCR, DI segmentation, DI classification, DI layout analysis, *etc.* Indeed, the pixel-labeling phase is considered as the first major step in a pixel-based DIA workflow after the image pre-processing/enhancement.

4.4.1.1. Pre-processing and foreground pixel selection

First, a HDI is fed as input and is read as a gray-scale image. The extraction of texture information is processed on gray-scale DIs without introducing a binarization task. A binarization step is avoided because it causes a loss of information specifically the textural information. Then, to deal with pixels at image borders when computing texture features on the whole image, a border replication step is introduced.

In this work, our goal is to have an overview of the page content by finding homogeneous regions with similar textural content as easily, quickly and automatically as possible rather than a fine characterization. Thus, in order to reduce data cardinality and obtain a significant gain in the computation time and used memory, the texture descriptors are extracted only on the selected foreground pixels. It is worth noting that the foreground texture is more interesting to categorize the type of DI content.

Therefore, the textural descriptors are extracted only on the selected foreground pixels. The foreground pixel selection step is performed using a standard parameter-free binarization method, the Otsu's method, to retrieve only those pixels representing information of the foreground (noise, text, graphics, *etc.*) [320]. However, using of the Otsu's method is beyond the scope of this work, it has provided good results [200]. They used the Otsu's method to segment and extract the text regions from a DI. Shijian and Tan binarized DIs using the Otsu's global thresholding method to retrieve the character pixels and subsequently identify the scripts and languages of noisy and degraded DIs [321]. Several comparative studies of the segmentation text/background or binarization methods for degraded HDIs have been reviewed [322, 323]. These studies do not agree on the best method and none has been shown to be perfect and suitable for HDIs, even local binarization approaches. Using a global thresholding approach, the Otsu's method provides an adequate and fast means of binarization to retrieve only the foreground pixels and extract texture features from only the selected foreground pixels.

As an example, for a full historical page document (1965×2750 pixels), scanned at 300 dpi, the number of the selected foreground pixels is equal to 26086. Thus, the rate of the selected foreground pixels is over $\frac{1}{200}$ of a DI pixels.

4.4.1.2. Texture feature extraction and multi-scale analysis

The texture feature extraction is performed using the pixel-wise technique, *i.e.* by using analysis windows of varying sizes in order to adopt a multi-resolution/multi-scale approach (*cf.* Section 3.3.2.4). The pixel-wise technique is chosen since it gives more reliable values and ensures more

accurate determination of texture boundary, however it has a high demand in memory and computational time. Using a multi-scale approach in DIA [324, 325, 326, 327] and pyramid methods in image processing [328, 329], rich information (e.g. gray-level distribution) can be produced since we can perceive differently textural characteristics at varying scales.

Typically, the sizes of sliding windows vary from (16×16) to (256×256) in the existing pixel-based methods using a multi-scale analysis. However, the computation time is highly dependent on the resolution, size of the analyzed DI and number of the selected foreground pixels. As a matter of fact, in this work the sizes of sliding windows vary only from (16×16) to (128×128), because beyond the (128×128) size the step of the texture feature extraction would be both costly and time-consuming. In addition, using a large size of a sliding window misleads an observation with coarse texture expression. Hence, the optimal size of each sliding windows determined respecting a constructive compromise between the computation time and pixel-labeling quality (reliable measurement and texture boundary).

In this work, the textural descriptors are only extracted from the selected foreground pixels of the gray-scale DIs at four different sizes of rectangular overlapping processing windows: $((16 \times 16)$, (32×32), (64×64) and $(128 \times 128))$ to adopt a multi-scale approach. Figure 4.2 illustrates an example of the four different pre-defined sizes of sliding windows: (16×16), (32×32), (64×64) and (128×128), and it shows that each window provides additional information on the textural properties.

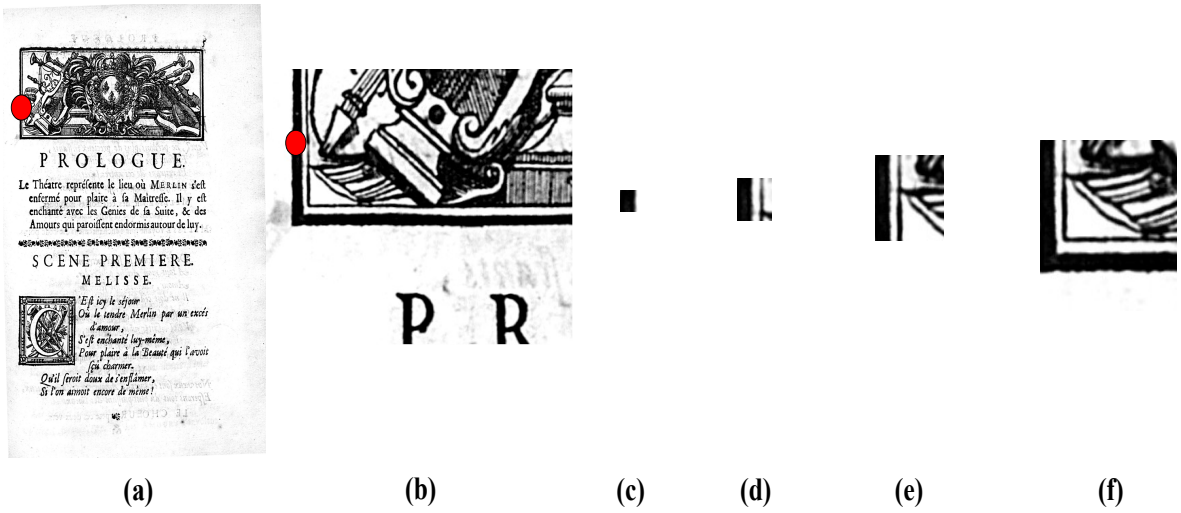


Figure 4.2.: Example of four different sizes of sliding windows. Figure (a) shows the original image with a selected pixel position. Figure (b) depicts an image zoom. Figures (c), (d), (e) and (f) illustrate (16×16), (32×32), (64×64) and (128×128) windows.

In this work, texture features are computed for analysis windows of four different sizes in order to adopt a multi-scale approach. The sliding window is shifted horizontally and vertically to scan the entire image. Therefore, a feature vector is computed on a foreground pixel-per-pixel basis. Each pixel is represented by scalar features, determined according to a small region bounded by contour of the analyzed sliding window. The analyzed sliding window is centered on that pixel. Subsequently, the extracted textural indices for the selected foreground pixels are aggregated into the N^f-dimensional (N^f-D) array on pixel-by-pixel basis, where N^f represents the number of extracted textural indices by applying multi-scale analysis.

4.4.1.3. Pixel-clustering and labeling

Since the texture feature extraction phase has been performed, we need to characterize the content of HDIs. The goal of this step is to structure the texture feature space within a hierarchical or

partitioning clustering technique in order to group pixels sharing similar characteristics and to identify and characterize unlabeled data (obtained from the texture feature extraction phase). The partition and analysis task of the set of unlabeled data into groups or clusters is necessary to segment the analyzed DI into regions which have homogeneous characteristics and similar properties with respect to the extracted texture features. This task is considered as a feature space structuring technique. Section A.1 in Appendix A presents briefly the different feature space structuring techniques proposed in the literature. The different feature space structuring techniques that have been used with HDIs are summarized in Table A.1. For instance, Nguyen *et al.* focused their study on specific graphics called drop caps and particularly on the extraction of shapes in these graphics, as part of an attempt to provide wider access to historical collections [330]. They found interesting classification results which were obtained by performing the hierarchical agglomerative clustering (HAC) algorithm on the stroke features of drop caps.

Since we opt for an unsupervised pixel-labeling scheme for comparing texture features, in this work we just need an unsupervised clustering step to group pixels sharing similar characteristics. Two conventional clustering techniques, k-means [331] and HAC [332], are chosen in the proposed pixel-labeling scheme for comparing texture features.

The k-means algorithm partitions the data samples into k clusters by using the squared Euclidean distance (SED) [333]. The $SED(x)$ of two multi-variate vectors $x = (x_1, x_2, \ldots, x_{N^f})^T$ and $y = (y_1, y_2, \ldots, y_{N^f})^T$ is defined as:

$$SED(x, y) = \sum_{i=0}^{N^f} (y_i - x_i)^2 \tag{4.1}$$

where N^f denotes the number of extracted textural indices per each selected foreground pixel by applying multi-scale analysis.

The HAC algorithm process consists in successively merging pairs of existing clusters where at each cluster grouping step, the choice of cluster pairs depends on the smallest distance, *i.e.* clusters are grouped if the intra-cluster inertia is minimal. Lai *et al.* [334] stated that the distance computed by the Ward method [335] gave the best results with the HAC method for content-based indexing of large image databases. Thus, the linkage between clusters is performed using the Ward criterion along with the weighted Euclidean distance ($WED(a, b)$) in the HAC algorithm [333]. The WED is defined as:

$$WED(a, b) = \sqrt{n_a n_b \frac{\sum_{k=1}^{N^f} \frac{1}{N^f} \|\overline{x_{ak}} - \overline{x_{bk}}\|}{n_a + n_b}} \tag{4.2}$$

where $\overline{x_{ak}} = \frac{\sum_{i=1}^{n_a} x_{ai}}{n_a}$ (resp. $\overline{x_{bk}} = \frac{\sum_{i=1}^{n_b} x_{bi}}{n_b}$) is the centroid of cluster a (resp. b) and n_a (resp. n_b) is the number of elements in cluster a (resp. b). N^f is the number of the vector features. The greater the WED (equation 4.2) between two clusters, corresponding to two different kinds of texture, the better the discrimination of the textural characteristics. The texture feature vectors computed at the selected foreground pixels are not identical and generate k clusters in the N^f-D feature space.

Therefore, in this work each pixel is automatically assigned to one of a number of possible clusters according to the contents of its feature vector by applying the HAC algorithm on the normalized textural features and setting the maximum number of homogeneous and similar content regions equal to the one defined in the ground-truth. The texture feature vectors are normalized to zero mean and unit standard deviation in order to avoid a domination of the higher numerical range of a few features. By partitioning texture-based feature vector sets into compact and well-separated clusters in the feature space, individual pixels are labeled without taking into account the spatial coordinates which lead to the application of the pixel-clustering and labeling steps, producing a pixel-labeled image as output. As a matter of fact, the spatial information is also not integrated

in the pixel-labeling scheme for comparing texture features, to avoid bias caused by introducing a refinement pixel-labeling phase with taking into consideration the topological relationships of pixels. In addition, the number of homogeneous and similar content regions has been set to the one defined in the ground-truth when performing the two conventional clustering techniques, k-means and HAC, in the pixel-labeling scheme for comparing texture features. The aim is to avoid inconsistencies and bias in assessments caused by estimating automatically the number of homogeneous and similar content regions and subsequently to ensure an objective understanding of the behavior of the evaluated texture feature sets.

4.4.2. Corpus and preparation of ground-truth

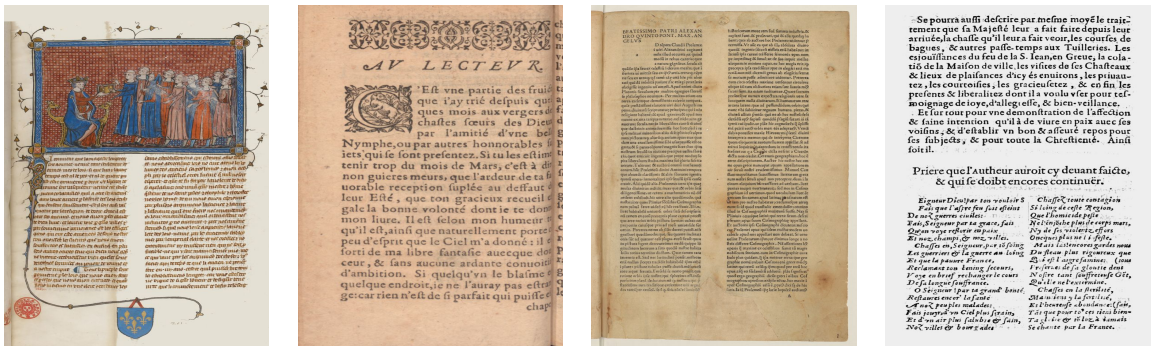
Although the issues of the realistic dataset availability and broadband access to researchers for the performance evaluation of contemporary DIs have been discussed and solved by Antonacopoulos *et al.* [336], representative datasets of HDIs are still hard to collect from several libraries. Then, defining the associated ground-truth of HDI corpus is still not a straightforward task due to their characteristics (e.g. page skew, superimposition of information layers, such as stamps, handwritten notes, noise, back-to-front interference). These characteristics complicate the definition of the appropriate and objective ground-truth, the characterization or segmentation of HDIs and make the processing of this kind of DIs a difficult task (*cf.* Figure 3.2).

Antonacopoulos *et al.* [336] considered a dataset as a good one if it is realistic (*i.e.* it must composed of real digitized DIs), comprehensive (*i.e.* it must well characterized and detailed for ensuring in-depth evaluation) and flexibly structured (*i.e.* to facilitate a selection of sub-sets with specific conditions). Thus, in our experiments, we focus on real scanned HDIs. The characteristics of our experimental corpus of HDIs are primarily: strong heterogeneity, with differences in layout, typography, illustration style, historic fonts, complex layouts (e.g. dense printing, irregular spacing, varying text column widths, marginal notes), ink shining through and historical spelling variants, *etc.* In addition to this specificity, the issues affecting DI layout analysis, such as the degradation properties (e.g. yellow pages, ink stains, back-to-front interference) and scanning defects (e.g. defects of curvature and light) are adequately covered.

The first experimental corpus used in this work which is called the *"DIGIDOC-Texture dataset"*, contains 1000 ground-truthed one-page HDIs which have been collected from Gallica[3], encompassing six centuries (1200-1900) of French history. The HDIs of the *"DIGIDOC-Texture dataset"* have been selected from several printed monographs and manuscripts across a variety of disciplines, such as novels, law texts, educational books (e.g. history, geography, nature) and xylographic booklets, to provide a broader range of HDI contents. They are gray-scale/color DIs which have been digitized at 300/400 dpi and saved in the TIFF format which provides a high resolution of digitized images.

The *"DIGIDOC-Texture dataset"* has been structured into four categories of real scanned HDIs differentiated by their content (*cf.* Figure 4.3), reflecting the challenges of this work to determine which texture features can be more adequate for segmenting the graphical regions from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand. The *"DIGIDOC-Texture dataset"* includes a sufficient number of images with both simple and complex layouts for each category of HDIs which have been ground-truthed to ensure a better understanding of the behavior of the evaluated texture feature sets. It is composed of:

- 250 pages containing graphics and one text font (*cf.* Figure 4.3(a)),
- 250 pages containing graphics and text with two different fonts (*cf.* Figure 4.3(b)),
- 250 pages containing only two fonts (*cf.* Figure 4.3(c)),
- 250 pages containing only three fonts (*cf.* Figure 4.3(d)).



(a) Graphics and one text font

(b) Graphics and text with two different fonts

(c) Only two fonts

(d) Only three fonts

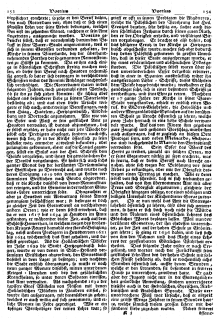
Figure 4.3.: HDI examples of the *"DIGIDOC-Texture dataset"* which have been collected from Gallica[3].

As part of the improving access to text (IMPACT)[1] project (an EU FP7 research project) and in the context of ICDAR conference and HIP workshop (2011 and 2013), 100 images were selected for historical document layout analysis and HBR competitions [39, 225]. This dataset (called in this work the *"HBR2013 dataset"*) which is used in different ICDAR competitions, has firstly the drawback to be limited (*i.e.* it contains only 100 pages and the ground-truth is not provided for all images (only six pages)). Secondly, it had been selected as it has as little as possible artifacts (e.g. severe page curl, arbitrary warping) to overcome the use of a separate image enhancement step before the DI layout analysis task. In addition, these competitions are related to HDI layout analysis and not to an end-to-end workflow. In addition, the *"HBR2013 dataset"* is composed of several binary images. Moreover, few images had been digitized at low resolution, that might potentially introduce a bias in the texture feature extraction and analysis tasks (*cf.* Figure 4.4(a)). Moreover, few images of the *"HBR2013 dataset"* have copyright notices at bottom of pages which may introduce an artificial information, thereafter inducing segmentation and characterization errors (*cf.* Figure 4.4(b)).

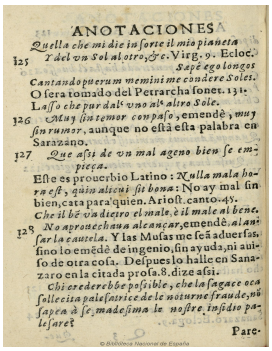
To study the scalability of the nine evaluated texture-based feature sets in a "public" dataset, experiments have been also carried out on the *"HBR2013 dataset"* which have been provided in the context ICDAR/HIP-HBR, a competition on the HBR, by the "Centre of competence in digitisation"[2] IMPACT research team. The *"HBR2013 dataset"* is composed of 100 binary, gray-scale or color HDIs which have been digitized at 150/300 dpi. We have structured the *"HBR2013 dataset"* into nine different categories differentiated by their content (*cf.* Figure 4.5):

- 03 pages containing only one font (*cf.* Figure 4.5(a)),
- 17 pages containing only two fonts (*cf.* Figure 4.5(b)),
- 09 pages containing graphics and text with two different fonts (*cf.* Figure 4.5(c)),
- 20 pages containing only three fonts (*cf.* Figure 4.5(d)),
- 06 pages containing graphics and text with three different fonts (*cf.* Figure 4.5(e)),
- 11 pages containing only four fonts (*cf.* Figure 4.5(f)),
- 15 pages containing graphics and text with four different fonts (*cf.* Figure 4.5(g)),
- 05 pages containing only five fonts (*cf.* Figure 4.5(h)),

[1] http://impact-project.eu
[2] http://digitisation.eu



(a) Binary images



(b) Copyright notices

Figure 4.4.: Illustration of the limitations of the *"HBR2013 dataset"*. Figure (a) shows few examples of binary images, while Figure (b) depicts few images of the *"HBR2013 dataset"* which have copyright notices at bottom of pages (zoomed regions on the copyright notices at bottom of pages are also illustrated).

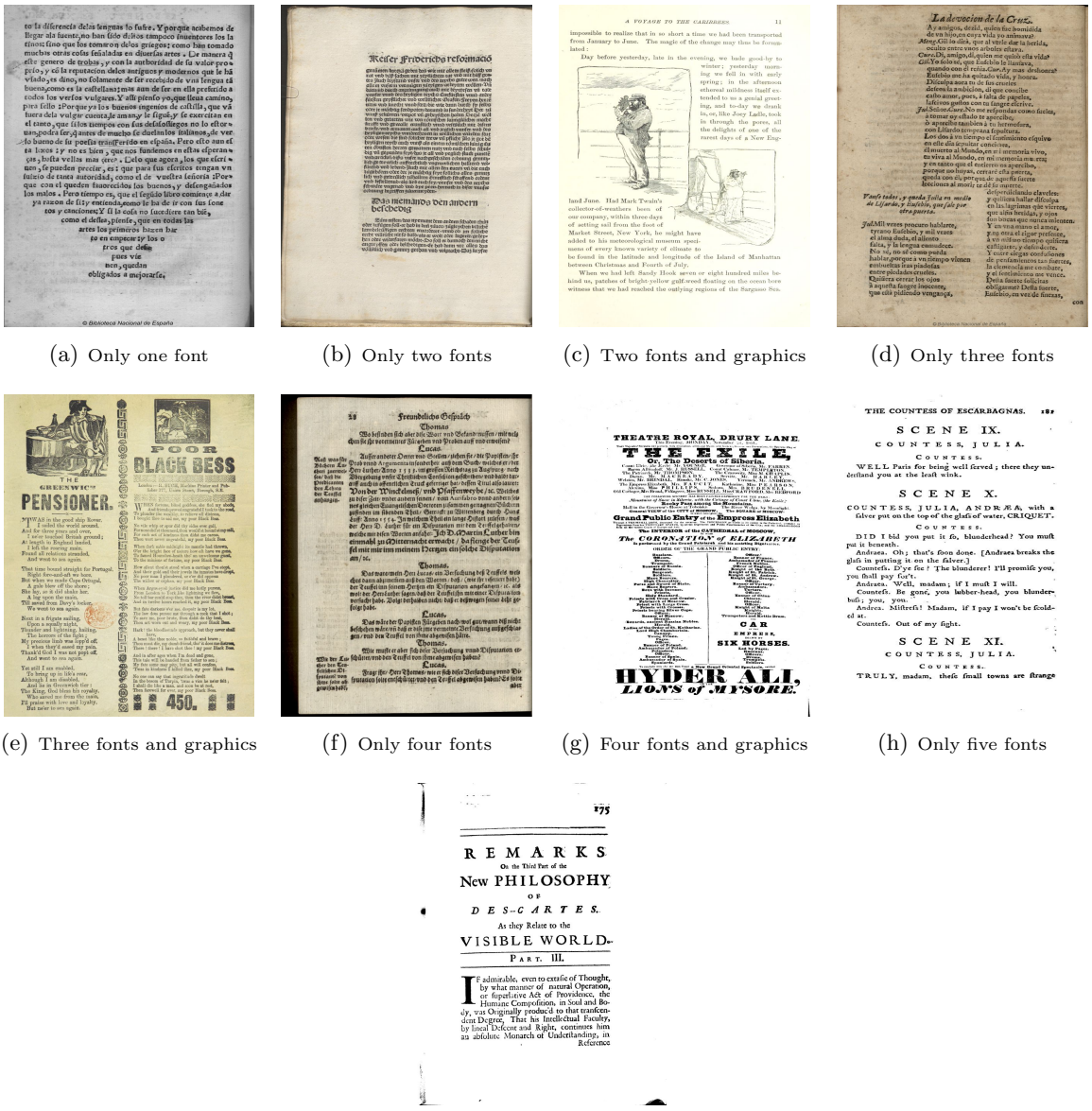
- 14 pages containing graphics and text with five different fonts (*cf.* Figure 4.5(i)).

Both for the DIGIDOC-Texture and HBR2013 datasets, the ground-truth for HDIs has been manually outlined using rectangular regions drawn around each selected zone. The regions have been ground-truthed by zoning each content type (*i.e.* each rectangular region has been classified into text or graphics). Different labels for regions with different fonts have been also defined for evaluating the performance of texture feature to separate various text fonts. Ground-truth has been performed using the ground-truthing editor, ground-truthing environment for document images (GEDI)[3], a public domain DI annotation tool that labels spatial boundaries of regions [337]. By specifying rectangular regions on a DI and assigning them to one of the many pre-defined content types, GEDI generates an XML schema representing the location on the page, height, width and label of each region (*cf.* Figure 4.6). The ground-truth has not been provided for all images of the *"HBR2013 dataset"* by the IMPACT research team (*i.e.* only six pages). Thus, the ground-truth of the *"HBR2013 dataset"* has been also carried out by using the GEDI tool.

4.4.3. Accuracy metrics for performance evaluation

Reed and DuBuf [142] stated that the fundamental questions of comparing texture-based methods are linked to how a comparative study can be carried out properly and how to evaluate their performance quantitatively. They classified the typical evaluation criteria into two categories: based on direct feature statistics and based on boundary accuracy after the segmentation task. In this work, we are not focus on an accurate pixel-based segmentation. We have narrowed our

[3] `http://gedigroundtruth.sourceforge.net/`



(a) Only one font (b) Only two fonts (c) Two fonts and graphics (d) Only three fonts

(e) Three fonts and graphics (f) Only four fonts (g) Four fonts and graphics (h) Only five fonts

(i) Five fonts and graphics

Figure 4.5.: HDI examples of the *"HBR2013 dataset"*.

focus to the use of the extracted low-level features to find homogeneous or similar content regions defined by similar textural indices and not on the basis of the state-of-the-art methods of grouping pixels according to the spatial relationships of pixels. Thus, we evaluate quantitatively the different analyzed texture features by computing various feature statistics which consists of the clustering and classification accuracy metrics. Section A.2 in Appendix A presents briefly the different clustering and classification accuracy metrics proposed in the literature. They are summarized in Table A.3.

The use of a set of objective evaluation criteria for a variety of DIA applications is considered as an open research topic [136]. Several DIA researchers have suggested initiating novel segmentation accuracy metrics. For instance, Yanikoglu and Vincent [338] suggested a complete environment which is called "Pink Panther", for creating automatically the segmentation ground-truth files and benchmarking different page segmentation algorithms. The segmentation quality of an algorithm is evaluated by comparing the segmentation output of the analyzed DI, described as a set of regions, to the corresponding previously created ground-truth through the error map. The error maps are used to quantify several kinds of errors (e.g. mis-classifications, splitting and merging of regions)

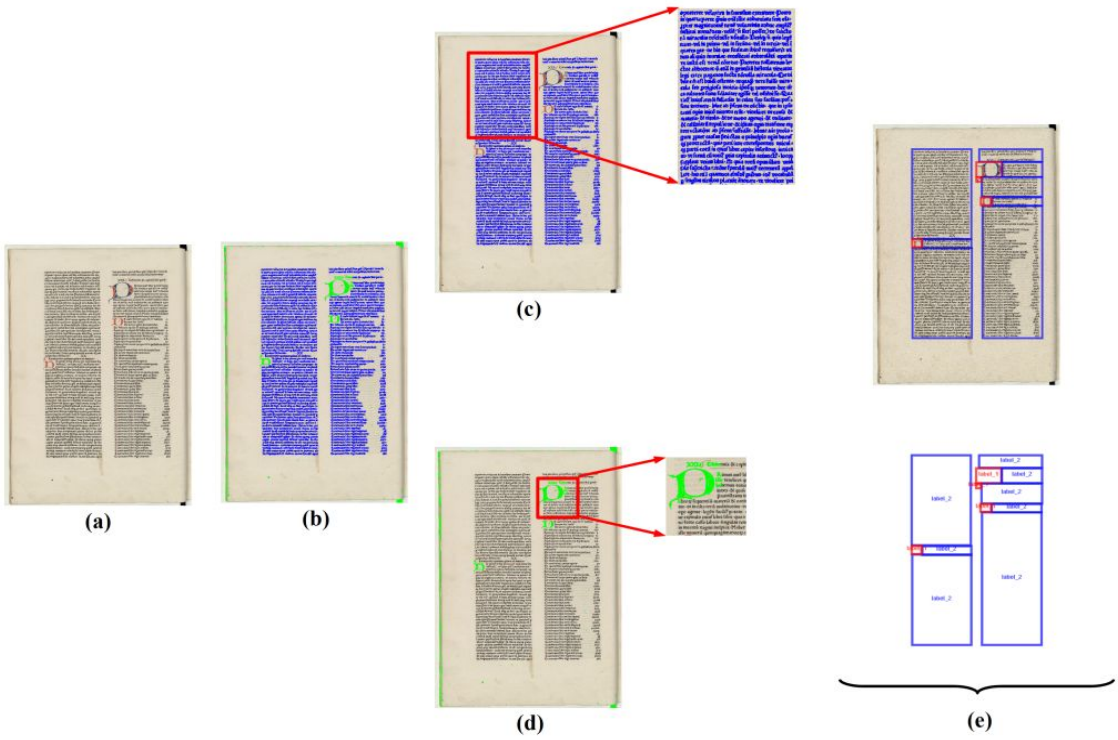


Figure 4.6.: Example of a pixel-labeling result. Figure (a) illustrates an original DI. Figure (b) shows final result of pixel-labeling task by analyzing the Gabor features. Figure (c) depicts a cluster representing the text, while Figure (d) shows a cluster representing the graphics. Figure (e) illustrates the associated ground-truth.

and to ensure a clear representation of all the errors associated with each pixel regardless the DI complexity.

Baird *et al.* [316] pointed out the zoning methodology problems and reported three accuracy metrics (per-pixel accuracy, per-page inventory accuracy and subjective segmentation quality) for a pixel-based approach evaluation. The per-pixel accuracy measures the fraction of all pixels in the DI that are correctly classified (*i.e.* pixels whose class label matches the pre-defined class in the ground-truth labels of the specified zones). The per-page inventory accuracy determines for each content class, the fraction of each page area that is classified as that class. Therefore, this metric analyzes the performance of queries for every content class by computing the precision and recall scores. The subjective segmentation quality provides a subjective assessment of the classification quality by using one of the following expressions: "good", "fair" or "poor". Baird *et al.* [316] reported that using rectangles in zoning can affect the per-pixel accuracy score due to the fact that some content can not be described by rectangular zones (e.g. handwritten regions) and due to the arbitrariness and inconsistency in zoning. Thus, they used the raw pixel-count data for determining what kind of errors the classifier encountered relative to the ground-truth particularly for the handwritten content. They also noted that using rectangles in zoning has an influence in computing the per-page inventory accuracy since the information of page layout is not included.

Vil'kin *et al.* [318] emphasized that the accuracy criterion and the set of test images for the clustering, classification and segmentation issues are in tremendous growth and continuous development. To compare the classification results of their proposed algorithm for DI segmentation, two criteria were used: percentage of correctly classified pixels measure (PPCM) and another criterion used on the ICDAR page segmentation competition 2007 which is called MatchScore [136]. The latter metric has the advantage to take into account the errors of small area. Silva [339] proposed two metrics: the completeness and purity, to evaluate the DIA performance applied specifically to

tables. The Jaccard coefficient (J) was used in the evaluation of a proposed pixel-based algorithm proposed by Hebert *et al.* [246] for the structure extraction from old newspapers. In their evaluation, the J metric measures a ratio between the number of correctly labeled pixels and the sum of pixels defined in the ground-truth. Ge *et al.* [340] evaluated their segmentation algorithm based on the detection and extraction of the salient objects in the images using the J metric.

The lack of the appropriate quantitative measures for the segmentation quality and the difficulty in defining criteria for specific application-dependent segmentation, are the shortcomings that limit researchers in an objective unsupervised evaluation of their results. For instance, the J metric is not suitable for assessing the accuracy of the proposed pixel-labeling scheme for comparing texture features because our work focuses on using the extracted low-level features to find homogeneous or similar content regions defined by similar textural indices. Given this objective, an external evaluation metric, the purity per block metric ($PPB(B, G^t)$) is defined in this work which evaluates the accuracy of a segmentation approach in terms of matching regions between the ground-truth and pixel-labeling regions. $PPB(B, G^t)$ is based on spatial overlaps of the ground-truth rectangle and the clustering result. It is defined as:

$$PPB(B, G^t) = \frac{1}{|G^t|} \sum_j \frac{1}{|\left\{ b_i \in g_j^t \right\}|} C_j \tag{4.3}$$

where

$$C_j = \max_{1 \leq k \leq k_{opt}} (|b_i, (b_i \in g_j^t) \wedge (l_{B_i} = k)|) \tag{4.4}$$

where $B = \{b_1, b_2, ..., b_i, ..., b_n\}$ and $G^t = \left\{ g_1^t, g_2^t, ..., g_j^t, ..., g_m^t \right\}$ are the sets of result blocks and rectangular regions of the ground-truth, respectively. $L_B = \{l_{B_1}, l_{B_2}, ..., l_{B_i}, ..., l_{B_n}\}$ corresponds to a set of labels obtained with the used pixel-clustering technique. b_i, g_j^t and l_{B_i} denote the result block, pre-defined rectangular region of the ground-truth and label corresponding to the result block obtained with the used pixel-clustering technique, respectively. $|.|$ is the number of pixels in a given block.

First, to evaluate quantitatively the different obtained results, the following clustering accuracy measures are computed in this work, silhouette width (SW), Jaccard coefficient (J) and purity per block (PPB).

- The ***silhouette width index (SW)*** measures the level of compactness and separation by analyzing the distribution of the observations into clusters [341].
- The ***Jaccard coefficient (J)*** is used to assess the similarity between the distributions of the observations in the clustering result and ground-truth. It represents the ratio of the number of pairs of data points which are clustered similarly in the clustering result and ground-truth [342].
- The ***purity per block (PPB)*** evaluates the pixel-labeling accuracy in terms of matching regions. It is based on spatial overlaps of the ground-truth rectangle and the pixel-labeling result to estimate the homogeneity/purity level per region.

Then, in order to provide an additional analysis and comparison with the computed clustering accuracy metrics and get an insight into the classification accuracy, a confusion matrix, error matrix or contingency table (M_c) is computed [343, 344]. From the M_c, several per-pixel classification accuracy metrics, including precision (P), recall (R), classification accuracy rate (CA) and F-score or F-measure (F) are performed in this work.

- The ***precision metric (P)*** corresponds to the proportion of the predicted cases that are correctly matched to the benchmark classifications. It is considered as a means of assessing the classification.

- The ***recall measure*** *(R)* indicates the proportion of real cases that are correctly predicted. It is considered a way to improve the classification.

- The ***classification accuracy rate*** *(CA)* metric corresponds to the ratio of the true classified predicted pixels and the total number of pixels.

- The ***F-measure*** *(F)* can be computed as a score resulting from the combination of the P and R accuracies by using a harmonic mean. It assesses both the homogeneity and the completeness criteria of a clustering result [345, 346, 347, 348, 349].

A detailed review of the used clustering and classification accuracy measures to evaluate the different extracted sets of texture feature has been conducted in Section A.2 and particularly in Appendix A.

4.5. Experiments and results

To analyze and evaluate the robustness of the nine investigated texture feature sets and provide additional insights into their classification accuracy and computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality), an informative benchmark of the performance and computational cost of each texture-based feature set is given in this section. Qualitative and numerical experiments are presented to demonstrate each texture-based feature set performance. Finally, based on the experimental results and observations, few recommendations about the choice of texture features which are firstly well suited for segmenting graphical regions from textual ones, discriminating text in a variety of situations of different fonts and scales and separating different types of graphics. Then, which texture features represent a constructive compromise between the pixel-labeling quality and computational cost. Finally, another set of experiments has been performed by using two different algorithms (k-means and HAC) in the pixel-clustering task (*cf.* Figure 4.1, Section 4.4.1.3) in order to compare their performance and to determine which one is more appropriate. As a consequence, we have divided the experiments into two parts:

1. Benchmarking of the nine extracted sets of texture feature based on using the HAC algorithm in the pixel-clustering step of the pixel-labeling scheme for comparing texture features (*cf.* Section 4.5.1).

2. Performance evaluation of the two different algorithms (k-means and HAC) in the pixel-clustering task of the pixel-labeling scheme for comparing texture features (*cf.* Section 4.5.2).

4.5.1. Benchmarking

The first experiment in this work proposes a comparative study of the nine investigated texture-based feature sets (Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor, Haar, Db3 and Db4), previously presented in Section 4.3 and detailed in Appendix B and particularly in Section B.1, using the proposed pixel-labeling scheme for comparing texture features (*cf.* Figure 4.1). First, we detail the computational cost by providing an additional insight into the computation time and complexity of each texture-based feature set is given (*cf.* Section 4.5.1.1). Qualitative and numerical experiments on the two datasets, DIGIDOC-Texture and HBR2013, are also given to demonstrate each texture-based feature set performance in Sections 4.5.1.2 and 4.5.1.3, respectively. A detailed analysis of the errors has been presented to show the limitations of a number of texture-based feature sets.

4.5.1.1. Computational cost

The benchmarking of the nine investigated texture-based approaches in this work has been run on a SGI Altix ICE 8200 cluster (1 central processing unit (CPU) and 2 gigabytes (GB) allocated memory on a Quad-Core $X5355@2.66GHz$ running Linux), without a very determined effort to achieve an optimized implementation of the investigated texture-based features. Analyzing the nine sets of texture descriptors from the DIGIDOC-Texture and HBR2013 datasets gives a total of 12150 analyzed DIs ($1000 + 250 + 100$ images $\times$ 9 different texture-based approaches). The "Two fonts and graphics" category of our HDI corpus is analyzed twice. First, every font in the text has a different label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 3 (graphics and text with two different fonts). Second, all fonts in the text have the same label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text). This distribution indicates out which texture features can be more suitable for segmenting a DI containing two text fonts and graphics into two/three classes, *i.e.* separating two different text fonts when a DI contains graphics.

The scalar features are extracted separately from the nine texture-based feature sets (Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor, Haar, Db3 and Db4) using four different sliding window sizes. Extracting each texture-based feature set by using a sliding window gives (*cf.* Sections 4.3 and B.1 and Table 4.4):

- ***16 Tamura indices*** (4 Tamura indices $\times$ 4 sliding window sizes for a multi-scale analysis): a 16-D feature vector which corresponds to the results of Tamura attribute extraction is assigned to every selected foreground pixel from the analyzed digitized DI.

- ***40 LBP indices*** (10 $LBP^{riu2}_{P_l=8,R_l=1}$ indices $\times$ 4 sliding window sizes): a 40-D feature vector is generated for each selected foreground pixel.

- ***176 GLRLM indices*** (44 GLRM indices $\times$ 4 sliding window sizes): 11 GLRLM indices are extracted for each scan direction. Thus, a total of 44 GLRM indices for four selected scan directions. A 176-D feature vector is generated for each selected foreground pixel.

- ***20 Auto-correlation indices*** (5 auto-correlation indices $\times$ 4 sliding window sizes): a 20-D feature vector which corresponds to the results of the auto-correlation attribute extraction, is associated to every selected foreground pixel from the digitized DI.

- ***72 GLCM indices*** (18 GLCM indices $\times$ 4 sliding window sizes): first, 16 GLCM features are extracted for two pre-defined distances (8 indices for each distance). In addition to the 16 extracted GLCM features, 2 other descriptors are computed for the two combined distances. Therefore, a 72-D feature vector which corresponds to the results of the GLCM attribute extraction, is associated to every selected foreground pixel from the digitized DI.

- ***192 Gabor indices*** (48 Gabor indices $\times$ 4 sliding window sizes): when convolving a DI with 24 Gabor channels (obtained by using 6 different spatial frequencies and 4 different orientations), 24 responses of filtered images or Gabor responses are generated. A feature vector (with dimension 48 to represent 24 channels) is produced per foreground pixel based on the computed mean and standard deviation of the magnitude response of the transformed analyzed image by the selective GF. Thus, a total of 48 Gabor indices are extracted from each selected foreground pixel defined in the analyzed sliding window. A 192-D feature vector is subsequently formed.

- ***80 Haar indices*** (20 Haar indices $\times$ 4 sliding window sizes): 2 indices for each Haar wavelet sub-band are extracted to form feature vector of 20 terms (10 sub-bands). Therefore, a 80-D feature vector is formed.

- ***80 Db3 indices*** (20 Db3 indices × 4 sliding window sizes): 2 indices for each Db3 wavelet sub-band are extracted to form feature vector of 20 terms (10 sub-bands). Therefore, a 80-D feature vector is formed.
- ***80 Db4 indices*** (20 Db4 indices × 4 sliding window sizes): 2 indices for each Db4 wavelet sub-band are extracted to form feature vector of 20 terms (10 sub-bands). Therefore, a 80-D feature vector is formed.

Nevertheless, there is awareness that maybe there are redundant and non-relevant indices when extracting each set of texture features with multi-scale analysis. As a matter of fact, a feature selection step can help select relevant features and remove redundant ones. However, in this work we are interested in raising issues related only to how these texture-based sets are compared with each other. We avoid bias caused by introducing a feature selection task, such as the methods based on the dimension reduction technique. Moreover, few rules and heuristics can usually be deduced when applying a feature selection task on the extracted texture features from a HDI/DHB. For instance, Abedi *et al.* [350] proposed a hybrid heuristic/random strategy for searching the optimal solution, based on evolutionary algorithms and heuristic methods. As a consequence, these deduced rules and heuristics can not be applied on another HDI/DHB due to the large variability of HDI/DHB contents. Hence, it is quite certain that a feature selection task can not be adapted to all kinds of HDIs/DHBs since the texture indices can have different ranges from a HDI/DHB to another one. Thus, a feature selection step has been avoided in this work.

Table 4.1.: A summary of the analyzed texture features in this work.

Features	**Description**
A- Tamura	
Coarseness (*cf.* equation (B.4))	This feature illustrates the scale and repetition rates of texture. Specifically, it measures the largest size at which a texture exists.
Contrast (*cf.* equation (B.5))	This descriptor measures the dynamic range of gray-levels in an image with taking into consideration the distribution polarization of black and white pixels.
Number of orientations (*cf.* equation (B.11))	This feature describes the local edge density and distribution of a texture.
Directionality (*cf.* equation (B.12))	This descriptor provides an insight into the global texture property over a region by measuring the total degree of texture directionality.
B- LBP	
Heights of the uniform bins of the histogram of binary patterns (*cf.* equation B.18)	These features represent the uniform patterns.
Height of the non-uniform bin of the histogram of binary patterns (*cf.* equation B.19)	This descriptor characterizes all the non-uniform patterns.
C- GLRLM	
Short-run emphasis (SRE) (*cf.* equation B.21)	This metric ensures the characterization of fine-grained textures by emphasizing short runs.
Long-run emphasis (LRE) (*cf.* equation B.22)	This feature helps to characterize textures with large homogeneous areas or coarse textures by emphasizing long runs.

Table 4.1 – continued from previous page

Features	Description
Low gray-level emphasis (LGRE) (*cf.* equation B.23)	This measure is orthogonal to SRE (*cf.* equation B.21) and it provides an insight of the dominance of many runs of low gray-level value in the analyzed texture.
High gray-level emphasis (HGRE) (*cf.* equation B.24)	This measure is orthogonal to LRE (*cf.* equation B.22) and it provides information on the dominance of many runs of high gray-level value in the analyzed texture.
Gray-level non-uniformity (GLNU) (*cf.* equation B.25)	This metric is focused on detecting the gray-level outliers from the histogram.
Run-length non-uniformity (RLNU) (*cf.* equation B.26)	This metric is an indicator of few run-length outliers which are dominating the histogram.
Run percentage (RPC) (*cf.* equation B.27)	This metric gives a glimpse into the overall histogram homogeneity. The maximum RPC value corresponds to the case where all runs are equal to the unity length regardless of the gray-level values.
Short-run low gray-level emphasis (SRLGE) (*cf.* equation B.28)	This measure is a combination of the two metrics: SRE (*cf.* equation B.21) and LGRE (*cf.* equation B.23) which estimates the dominance of many short runs of low gray-level value.
Long-run high gray-level emphasis (LRHGE) (*cf.* equation B.29)	This feature is the complementary metric to SRLGE (*cf.* equation B.28). It characterizes the combination of long high gray-level value runs.
Short-run high gray-level emphasis (SRHGE) (*cf.* equation B.30)	This measure is both orthogonal to SRLGE (*cf.* equation B.28) and LRHGE (*cf.* equation B.29). It carries out the domination of short runs with high intensity gray-levels in the analyzed texture.
Long-run low gray-level emphasis (LRLGE) (*cf.* equation B.31)	This feature is the complementary metric to SRHGE (*cf.* equation B.30). It allows to characterize long runs with low intensity gray-levels in the analyzed texture.
D- Auto-correlation	
Main angle of the rose of directions (*cf.* equation (B.35))	This metric ensures the characterization of the main orientation of a texture.
Intensity of the auto-correlation function for the main orientation (*cf.* equation (B.36))	This feature helps to characterize the anisotropy of a texture.
Variance of the intensities of the rose of directions (*cf.* equation (B.37))	This measure provides an insight of the overall shape of the rose of directions.
Mean stroke width along specific directions (*cf.* Algorithm 8)	This measure estimates the mean stroke width along specific directions.
Mean stroke height along specific directions (*cf.* Algorithm 9)	This metric corresponds to the estimation of mean stroke height along specific directions.

Table 4.1 – continued from previous page

Features	Description
E- GLCM	
Maximum probability (*cf.* equation (B.43))	This metric ensures the record of the highest GLCM element. High values of GLCM element will occurred if one combination of pixels dominates pixel pairs.
Correlation metric (*cf.* equation (B.44))	This feature helps to measure the gray-level linear dependence between pixels at the specified positions relative to each other. It has a large value when the values are uniformly distributed in the GLCM and a low value otherwise.
Energy (*cf.* equation (B.45))	This measure which has also been called angular second moment, provides an insight of image homogeneity. It has low value when the probabilities of the gray-level pairs have very similar values and a high value otherwise.
Entropy (*cf.* equation (B.46))	This metric characterizes the energy values for pixel combinations. It measures the disorder or randomness of the GLCM. Inhomogeneous texture have low first order entropy, while a homogeneous texture has a high entropy.
Contrast (*cf.* equation (B.47))	This metric which has also been called inertia, corresponds to a measure of the contrast by computing a difference moment of the GLCM and it estimates the contrast or it quantifies local variation present in the analyzed image.
Local homogeneity (*cf.* equation (B.48))	This measure has also been called inverse difference moment. It is higher when we find the same pair of pixels which is in the case that the gray-level is uniform or when there is a spatial periodicity.
Cluster shade (*cf.* equation (B.49))	This metric corresponds to a measure of the gray-level distribution around the mean, with a high ability to discriminate the third order. It measures the skewness of the GLCM (*i.e.* lack of symmetry). When it is high, the analyzed image is not symmetric.
Cluster prominence (*cf.* equation (B.50))	This metric corresponds to a measure of the gray-level distribution around the mean, with a high ability to discriminate the fourth order. It also measures the skewness of the GLCM.
Energy mean (*cf.* equation (B.51))	This metric corresponds to the mean of the energy feature computed from the two distance values $d_c = 1, 2$.
Energy standard deviation (*cf.* equation (B.52))	This metric corresponds to the standard deviation of the energy feature computed from the two distance values $d_c = 1, 2$. It characterizes the uniformity of the texture when varying the specified distance.
F- Gabor	
Mean of the Gabor filtered magnitude responses (*cf.* equation (B.54))	This feature characterizes the average of the Gabor filtered magnitude response corresponding to all pixels defined in the analyzed sliding window of the filtered image. This descriptor quantifies how the dominant texture properties of the analyzed image match to the set of spatial-frequency components of the fixed GF.

Table 4.1 – continued from previous page

Features	Description
Standard deviation of the Gabor filtered magnitude response (*cf.* equation (B.55))	This descriptor determines how much the dispersion from the computed mean of the Gabor filtered magnitude response exists.
G- Wavelet (Haar, Db3 and Db4)	
Mean of the wavelet transform coefficients (*cf.* equation (B.60))	This feature characterizes the average of the wavelet transform coefficients for each sub-band defined in the analyzed sliding window of the image. This descriptor represents the average of 2-D signal in various frequency bands.
Standard deviation of the wavelet transform coefficients (*cf.* equation (B.61))	This descriptor determines how much the dispersion from the computed mean of wavelet transform coefficients exists.

It is worth noting that the code implemented for the texture feature analysis task is not optimized in this work. An optimization process by using the single instruction, multiple data (SIMD) parallelization on different general-purpose processing on graphics processing units (GPGPU) graphics cards is especially necessary to quickly assess the nine investigated texture feature sets and have better computational cost.

An additional insight into the computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality) is provided in Table 4.4. The processing time depends highly on the resolution, size of the input image and number of the selected foreground pixels. An example of a full page document scanned at 300 dpi (1965×2750 pixels) is illustrated in Table 4.4. The highest time required to process a page (1965×2750 pixels) is obtained when using the wavelet approaches while the lowest one is obtained when using the GLCM descriptors (*i.e.* it is reduced to only 14 seconds). The computation time of each texture feature sets is in concordance with the complexity. We can see that the Db4-based approach has the highest complexity while the lowest one is noted for the GLCM-based approach (*cf.* Table 4.4). Therefore, this study states the GLCM-based approach as the best one in terms of processing time and complexity. However, the GLCM and Gabor-based approaches are the highest memory-consuming (*i.e.* more than 587MB used memory). We note that even the three investigated wavelets consume a similar amount of memory, they have different computation times. The Haar-based approach is the best one among the three investigated wavelets in terms of the computational cost. This confirms that the Haar wavelet transform is the fastest among the examined wavelets (*cf.* Section 4.3.7). However, the auto-correlation and LBP-based approach have similar computational cost, they have different feature dimensions (*i.e.* dimension of the LBP feature vector is the double of the auto-correlation one). Nevertheless, we observe the increase of the feature dimension of the Gabor and GLRLM-based approaches (*i.e.* Gabor and GLRLM signatures correspond to a set of vectors composed of 192 and 176 numerical values, respectively).

4.5.1.2. Qualitative results

A visual comparison of the resulting images using the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"DIGIDOC-Texture dataset"* is illustrated in Figures 4.8, 4.9, 4.10, 4.11 and 4.12. On the other side, the resulting images of using the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"HBR2013 dataset"* are depicted in Figures 4.14, 4.15, 4.16, 4.17, 4.18 and 4.19. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.

Measures of accuracy metrics are presented at the bottom of each image in Figures 4.8, 4.9, 4.10, 4.11, 4.12, 4.14, 4.15, 4.16, 4.17, 4.18 and 4.19.

By visual inspection of the obtained pixel-labeled HDIs, we note that most of the investigated texture-based approaches provide satisfying results particularly in distinguishing the textual regions from the graphical ones. We also observe that the Gabor-based approach performs considerably better in segmenting documents containing only textual regions with distinct fonts.

1. ***"DIGIDOC-Texture dataset"***
   Figures 4.8, 4.9, 4.10, 4.11 and 4.12 illustrate few examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* categories of HDIs from the *"DIGIDOC-Texture dataset"*, respectively. The *"Two fonts and graphics*"* category of HDIs represents the case when every font in the text has a distinct label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 3 (graphics and text with two different fonts). On the other side, the *"Two fonts and graphics**"* category of HDIs represents the case when all fonts in the text have the same label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text). This distribution points out which texture features can be more adequate for segmenting documents containing two text fonts and graphics into two/three classes, *i.e.* separating two distinct text fonts when the HDIs contain graphics.

   In Figure 4.8, where the analyzed HDI has a complex layout and contains one text font and graphics, the results given by analyzing the nine investigated texture-based feature sets on the proposed pixel-labeling scheme are relatively similar and satisfying in distinguishing the textual regions from the graphical ones when comparing visually the segmentation results. The pixel-labeling results of the nine extracted texture feature sets show a significant discriminating power for separating text (single font) and graphic regions. Nevertheless, by comparing the visual results given by analyzing the nine investigated texture-based feature sets on the proposed pixel-labeling scheme, we note that the graphic regions (green) are more homogeneous when using Gabor features (*cf.* Figure 4.8(f)) than when using the other texture features. However, the Gabor features have more difficulty separating textual regions (blue) when they are too spatially close to the graphical ones (*i.e.* textual regions which are spatially close to the graphic ones have been mis-labeled). On the other hand, in Figure 4.9, where the HDI under consideration contains two fonts and graphics, the nine investigated sets of texture features can not separate properly textual regions with different sizes and fonts. By analyzing the most sets of texture features for the *"Two fonts and graphics*"* category of HDIs, two clusters are produced for graphic regions by discriminating the noise on the HDI borders. This points out that the texture features have also more difficulty segmenting two distinct text fonts when the involved HDI contains graphics.

   We also observe that the wavelet-based approaches and more specifically Db3 and Db4, perform slightly similarly to the Gabor one (*cf.* Figure 4.7) and particularly in the case of HDIs containing graphics and text (*cf.* Figure 4.8). In certain cases however, the Gabor-based approach confuses the uppercase text and the graphical components (*cf.* Figures 4.7(a) and 4.7(b)) unlike the wavelet-based approach (*cf.* Figures 4.7(e) and 4.7(f)). This confusion can be explained by the limitations of the Gabor approach to separate spatially close distinct kinds of information (*i.e.* the vertical/horizontal spacing is too small). Indeed, the Gabor features are extracted for a specified range of frequency and direction values. Thus, the performance of the Gabor approach depends directly on the layout document. However, when using the Gabor primitives, we can see that distinct kinds of graphics can be discriminated (*cf.* Figures 4.7(c) and 4.7(d)).

   Then, by analyzing the visual results of the *"Two fonts and graphics**"* category of HDIs (*cf.* Figure 4.10), we observe that the GLCM descriptors are much better for segmenting text and graphic regions (*cf.* Figure 4.10(e)). Moreover, we conclude that the investigated

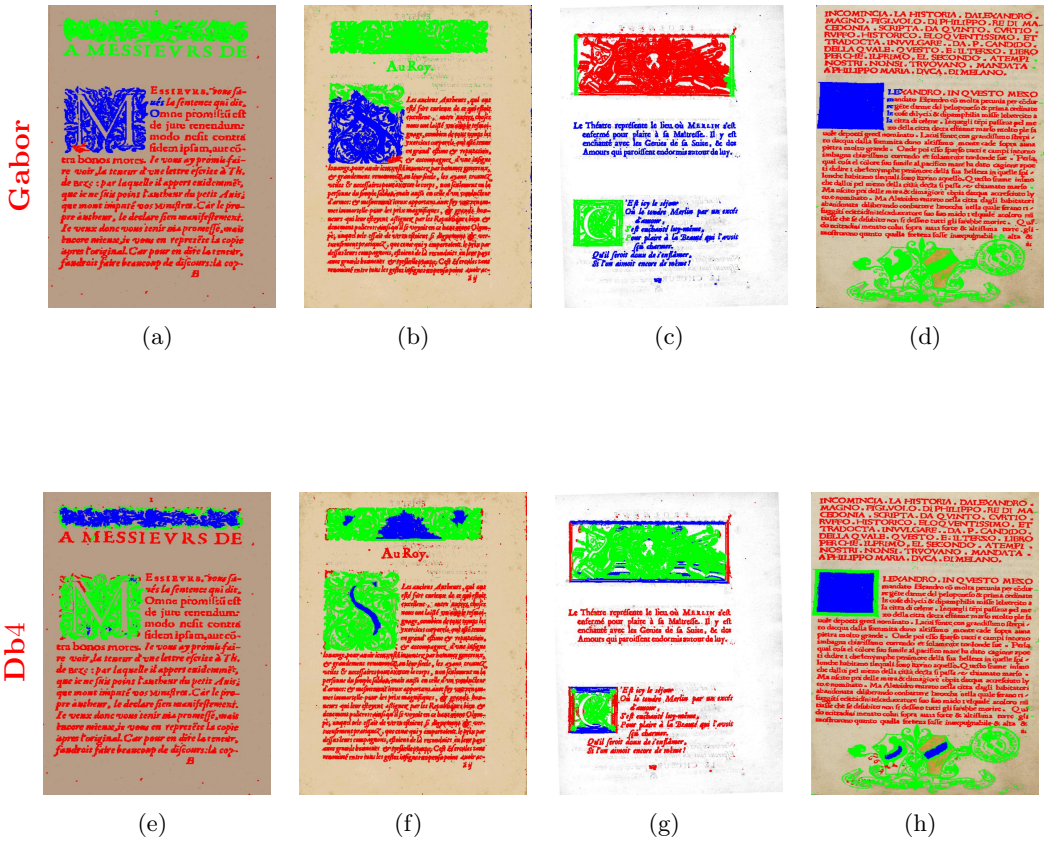
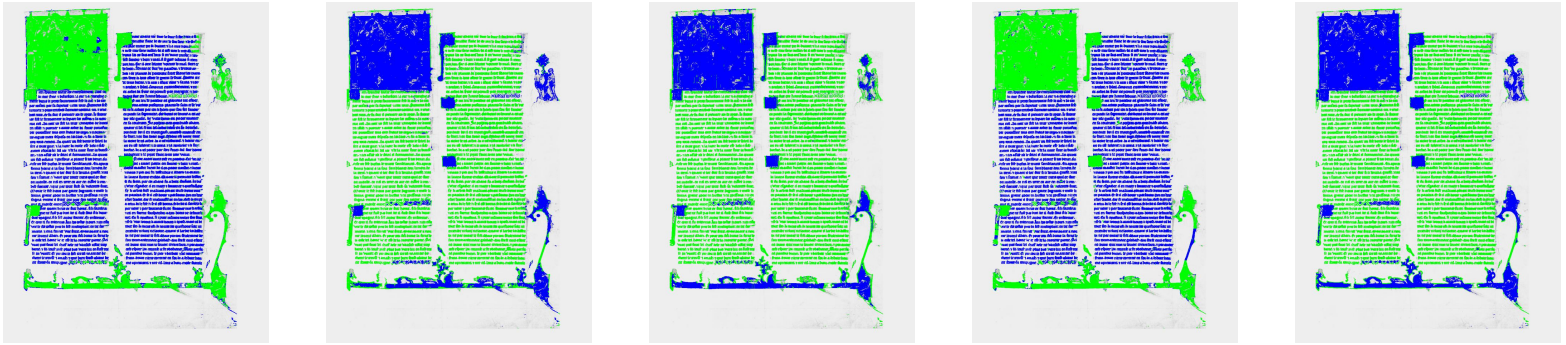


Figure 4.7.: Examples of resulting images of the proposed pixel-labeling scheme using the Gabor and Db4 features on the *"Two fonts and graphics** "* category of HDIs from the *"DIGIDOC-Texture dataset"*. Figures (a), (b), (c) and (d) show four resulting image examples of the *"Two fonts and graphics** "* category of HDIs from the *"DIGIDOC-Texture dataset"* using the Gabor features on the proposed pixel-labeling scheme. Figures (e), (f), (g) and (h) illustrate four resulting image examples of the *"Two fonts and graphics** "* category of HDIs from the *"DIGIDOC-Texture dataset"* using the Gabor features on the proposed pixel-labeling scheme. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.

texture feature are more suitable for segmenting documents containing two text fonts and graphics into two classes. This may raise questions about the importance of using recursive clustering methods in order to ensure the distinction between distinct text fonts and various graphic types. In a HDI example from *"DIGIDOC-Texture dataset"* which contains only two fonts, we observe that the GLCM-based (*cf.* Figure 4.11(e)) and Gabor-based (*cf.* Figure 4.11(f)) approaches provide the best visual results by distinguishing two different text fonts (handwritten notes in the margins and printed text). Finally, we demonstrate that the Gabor features are the best in segregating three different fonts, text with S_1^f size font (red), text with $S_2^f \neq S_1^f$ size font (blue) and italic (green) fonts in Figure 4.12(f). This may be confirmed by the frequent use of the Gabor descriptors mainly to identify script and language and for character and font recognition in the literature [285, 281], since the Gabor features are known to be sensitive to the stroke width. On the other side, for the other texture features including the three investigated kinds of wavelets (*cf.* Figures 4.12(g), 4.12(h) and 4.12(i)), the outcomes are poorer in segregating three different fonts.



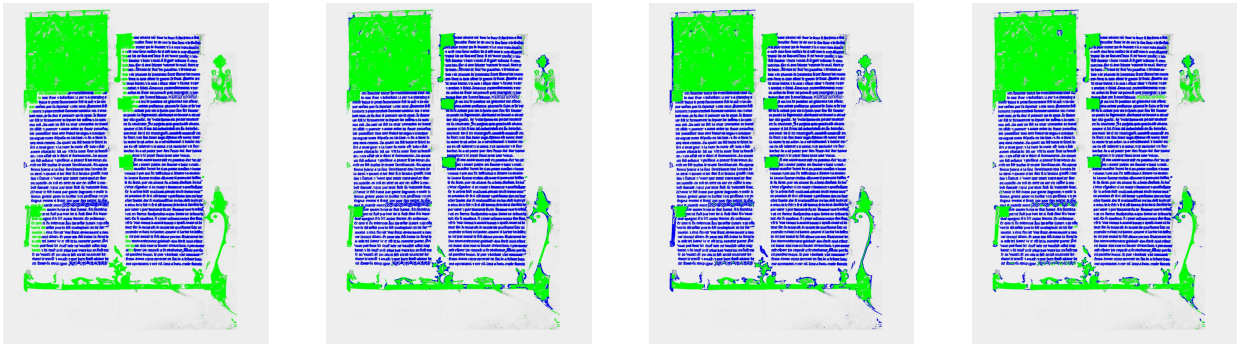
(a) ***Tamura***
$SW = 0.54$ $J = 0.86$
$PPB = 0.91$ $F = 0.96$

(b) ***LBP***
$SW = 0.42$ $J = 0.78$
$PPB = 0.89$ $F = 0.94$

(c) ***GLRLM***
$SW = 0.40$ $J = 0.73$
$PPB = 0.87$ $F = 0.92$

(d) ***Auto-correlation***
$SW = 0.72$ $J = 0.88$
$PPB = 0.92$ $F = 0.97$

(e) ***GLCM***
$SW = 0.69$ $J = 0.82$
$PPB = 0.90$ $F = 0.95$



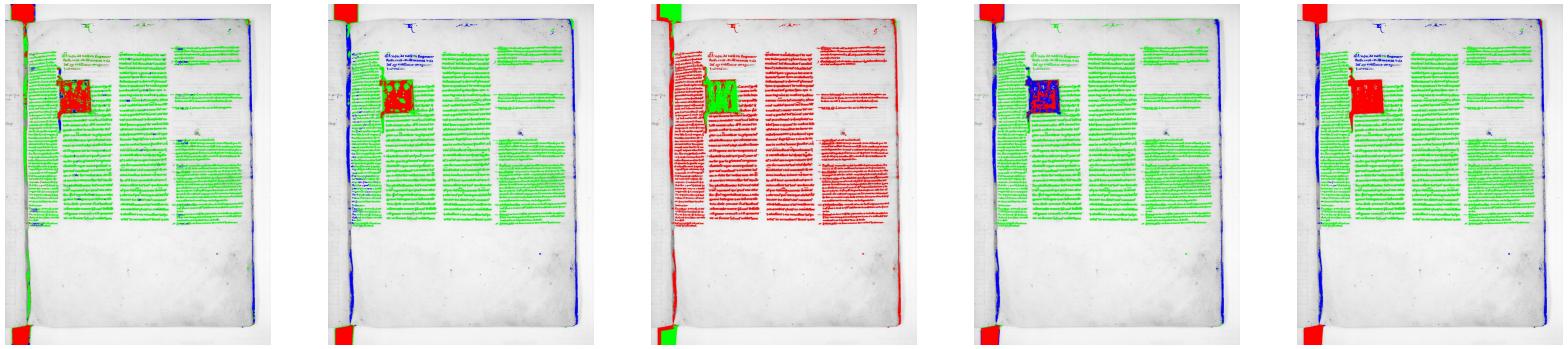
(f) ***Gabor***
$SW = 0.42$ $J = 0.87$
$PPB = 0.94$ $F = 0.96$

(g) ***Haar***
$SW = 0.61$ $J = 0.86$
$PPB = 0.94$ $F = 0.96$

(h) ***Db3***
$SW = 0.66$ $J = 0.81$
$PPB = 0.94$ $F = 0.95$

(i) ***Db4***
$SW = 0.61$ $J = 0.88$
$PPB = 0.92$ $F = 0.97$

Figure 4.8.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"One font and graphics"*** category of HDIs from the ***"DIGIDOC-Texture dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



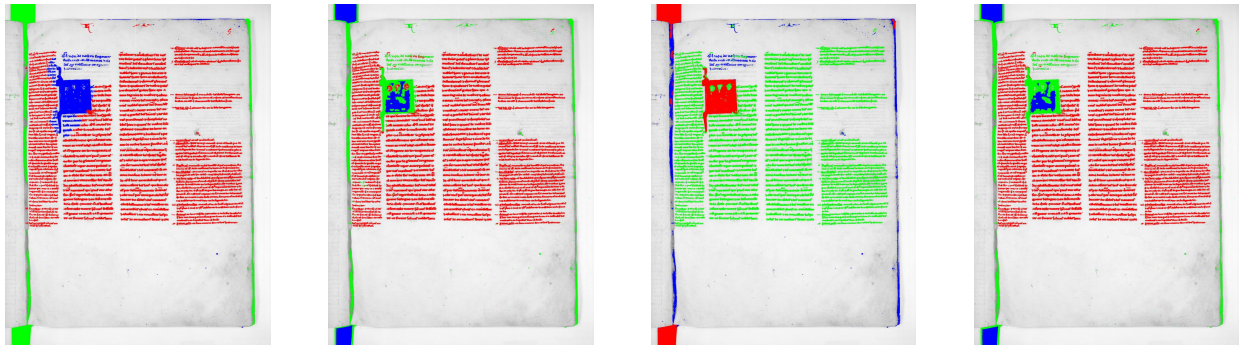
(a) ***Tamura***
$SW = 0.26$ $J = 0.46$
$PPB = 0.85$ $F = 0.57$

(b) ***LBP***
$SW = 0.75$ $J = 0.46$
$PPB = 0.89$ $F = 0.69$

(c) ***GLRLM***
$SW = 0.68$ $J = 0.46$
$PPB = 0.89$ $F = 0.52$

(d) ***Auto-correlation***
$SW = 0.62$ $J = 0.47$
$PPB = 0.85$ $F = 0.54$

(e) ***GLCM***
$SW = 0.39$ $J = 0.48$
$PPB = 0.93$ $F = 0.70$



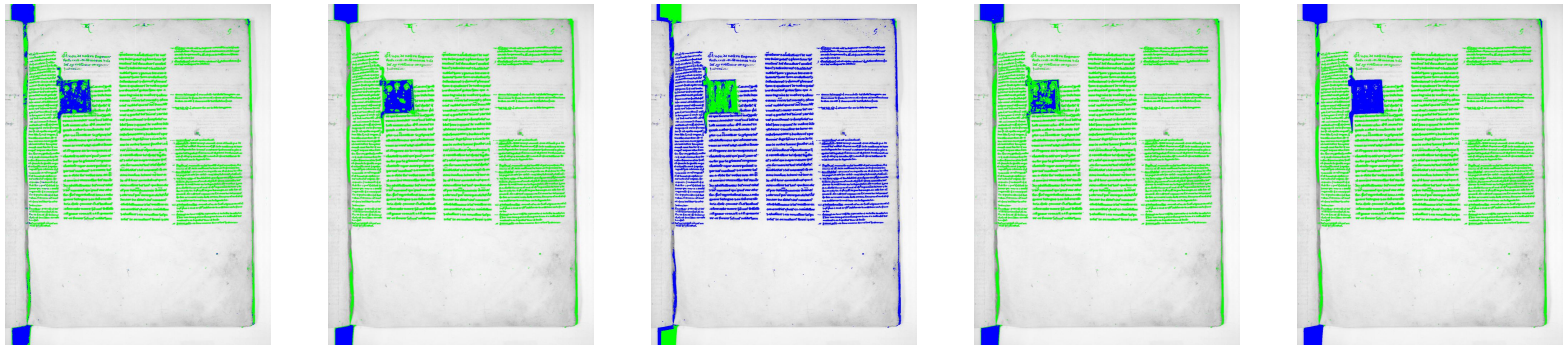
(f) ***Gabor***
$SW = 0.36$ $J = 0.47$
$PPB = 0.95$ $F = 0.72$

(g) ***Haar***
$SW = 0.59$ $J = 0.46$
$PPB = 0.87$ $F = 0.53$

(h) ***Db3***
$SW = 0.60$ $J = 0.48$
$PPB = 0.93$ $F = 0.69$

(i) ***Db4***
$SW = 0.59$ $J = 0.47$
$PPB = 0.90$ $F = 0.52$

Figure 4.9.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Two fonts and graphics*"*** category of HDIs from the ***"DIGIDOC-Texture dataset"***. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



(a) ***Tamura***
$SW = 0.27$ $J = 0.97$
$PPB = 0.88$ $F = 0.89$

(b) ***LBP***
$SW = 0.81$ $J = 0.99$
$PPB = 0.94$ $F = 0.90$

(c) ***GLRLM***
$SW = 0.68$ $J = 0.99$
$PPB = 0.89$ $F = 0.91$

(d) ***Auto-correlation***
$SW = 0.74$ $J = 0.99$
$PPB = 0.92$ $F = 0.81$

(e) ***GLCM***
$SW = 0.60$ $J = 0.99$
$PPB = 0.96$ $F = 0.98$

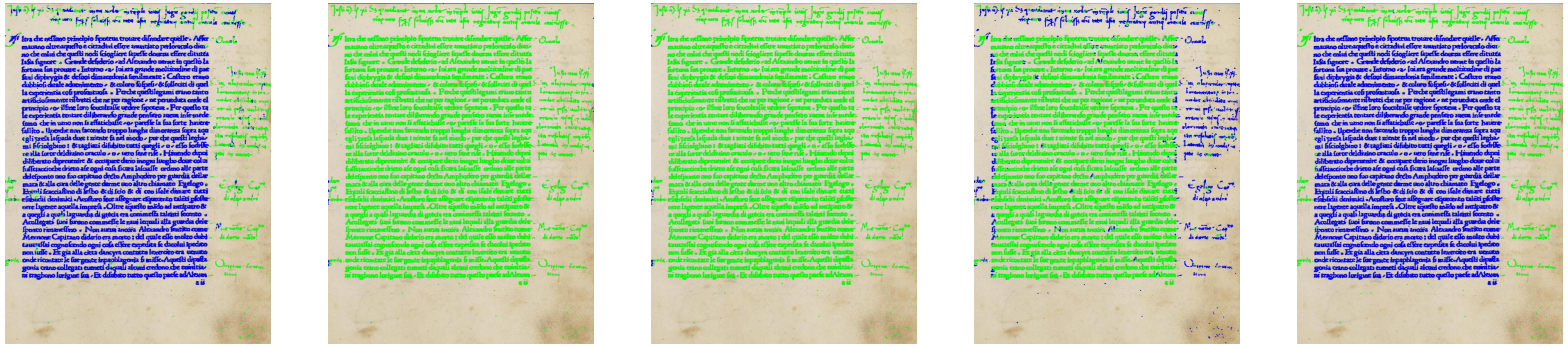
(f) ***Gabor***
$SW = 0.43$ $J = 0.85$
$PPB = 0.95$ $F = 0.89$

(g) ***Haar***
$SW = 0.66$ $J = 0.89$
$PPB = 0.92$ $F = 0.90$

(h) ***Db3***
$SW = 0.82$ $J = 0.98$
$PPB = 0.97$ $F = 0.97$

(i) ***Db4***
$SW = 0.67$ $J = 0.89$
$PPB = 0.95$ $F = 0.91$

Figure 4.10.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Two fonts and graphics**"*** category of HDIs from the ***"DIGIDOC-Texture dataset"***. *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text). Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



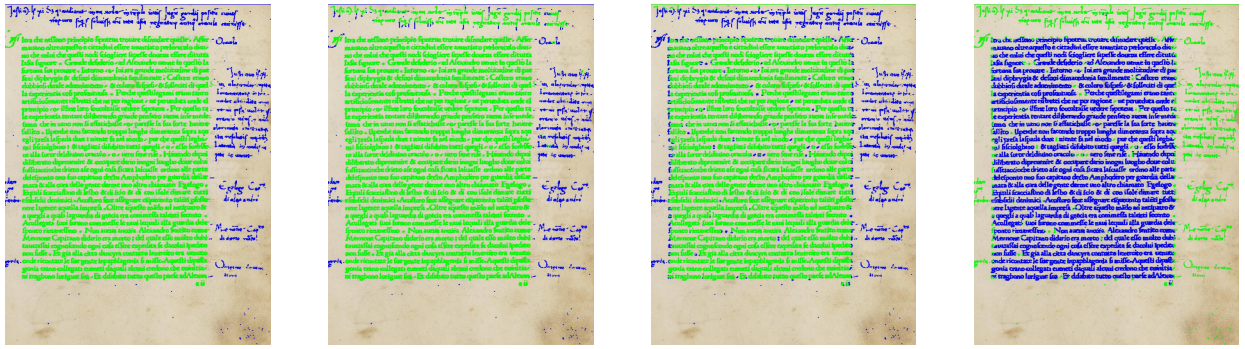
(a) ***Tamura***
$SW = 0.27 \quad J = 0.91$
$PPB = 0.85 \quad F = 0.90$

(b) ***LBP***
$SW = 0.94 \quad J = 0.80$
$PPB = 0.92 \quad F = 0.69$

(c) ***GLRLM***
$SW = 0.99 \quad J = 0.79$
$PPB = 0.92 \quad F = 0.66$

(d) ***Auto-correlation***
$SW = 0.72 \quad J = 0.88$
$PPB = 0.83 \quad F = 0.88$

(e) ***GLCM***
$SW = 0.62 \quad J = 0.99$
$PPB = 0.99 \quad F = 0.99$



(f) ***Gabor***
$SW = 0.41 \quad J = 0.99$
$PPB = 0.99 \quad F = 0.99$

(g) ***Haar***
$SW = 0.60 \quad J = 0.90$
$PPB = 0.92 \quad F = 0.89$

(h) ***Db3***
$SW = 0.54 \quad J = 0.90$
$PPB = 0.93 \quad F = 0.91$

(i) ***Db4***
$SW = 0.50 \quad J = 0.89$
$PPB = 0.96 \quad F = 0.91$

Figure 4.11.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Only two fonts"*** category of HDIs from the ***"DIGIDOC-Texture dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.

(a) ***Tamura***
$SW = 0.44 \quad J = 0.42$
$PPB = 0.75 \quad F = 0.55$

(b) ***LBP***
$SW = 0.12 \quad J = 0.60$
$PPB = 0.74 \quad F = 0.70$

(c) ***GLRLM***
$SW = 0.02 \quad J = 0.35$
$PPB = 0.79 \quad F = 0.38$

(d) ***Auto-correlation***
$SW = -0.13 \quad J = 0.46$
$PPB = 0.76 \quad F = 0.54$

(e) ***GLCM***
$SW = 0.05 \quad J = 0.33$
$PPB = 0.69 \quad F = 0.61$

(f) ***Gabor***
$SW = 0.36 \quad J = 1.00$
$PPB = 1.00 \quad F = 1.00$

(g) ***Haar***
$SW = 0.11 \quad J = 0.46$
$PPB = 0.72 \quad F = 0.69$

(h) ***Db3***
$SW = 0.13 \quad J = 0.37$
$PPB = 0.61 \quad F = 0.53$

(i) ***Db4***
$SW = 0.16 \quad J = 0.55$
$PPB = 0.75 \quad F = 0.64$

Figure 4.12.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Only three fonts"*** category of HDIs from the ***"DIGIDOC-Texture dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.

2. ***"HBR2013 dataset"***
   Figures 4.14, {4.15, 4.16}, 4.17, {4.18 and 4.19} illustrate few examples the resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"Only two fonts"*, *"Two fonts and graphics"*, *"Only three fonts"* and *"Three fonts and graphics"* categories of HDIs from the *"HBR2013 dataset"*, respectively.

   In the case of a HDI containing only textual regions with two different fonts (*cf.* Figure 4.14), we observe that the Gabor features are the best in segregating two different fonts, *i.e.* we distinguish two different text fonts, text with S_1^f size font (green) and text with $S_2^f \leq S_1^f$ size font (blue) (*cf.* Figure 4.14(f)). On the other side, the other investigated texture features have not borne the desired goal of segregating two different fonts. This strengthens our previous results obtained for the *"DIGIDOC-Texture dataset"* and confirms our assumption that the Gabor descriptors are the most suitable for font segmentation, since they are known to be sensitive to the stroke width. In figure 4.15, we see that the auto-correlation, Gabor and the three investigated wavelet-based approaches produce two clusters for graphic regions by discriminating many orientations that are present to different extents in graphic blocks. This confirms that these descriptors generally provide the main orientation of a texture. Moreover, this strengthens our previous observations deduced when analyzing the *"Two fonts and graphics"* category of HDIs in the *"DIGIDOC-Texture dataset"* the that these features have also more difficulty segmenting two distinct text fonts when the documents also contain graphics. We conclude that most investigated texture feature sets can not separate properly textual regions with different sizes and fonts and particularly when the documents also contain graphics. A suitable alternative is to use recursive clustering methods in order to ensure the distinction between distinct text fonts and various graphic types when the documents under consideration are complex and contains graphics and various kinds of fonts. Similarly, in Figure 4.16 all investigated sets of texture features can not separate properly textual regions with different sizes and fonts and particularly when the documents also contain graphics. When analyzing the Gabor and the three investigated wavelets, two clusters are produced for the graphic regions by discriminating the horizontal nets from the vertical black borders (noise generated during the digitization process) which should constitute a class on its own (*cf.* Figures 4.16(f), 4.16(g), 4.16(h) and 4.16(i)). This may raise questions about the defined ground-truth which is to a certain extent subjective and should consider the noise pixels in an another ground-truth class different of the already defined ones. Nevertheless, defining a pixel-based ground-truth in HDIs with taking account the noise pixels is not a straightforward task. In Figure 4.17, we observe that all investigated texture features even the Gabor features have failed to separate text fonts when the analyzed HDI contains only three different text fonts. This may be explained by the fact that the analyzed HDI has a copyright notice at the bottom of the page. This copyright notice has introduced an artificial texture information and subsequently a bias in the texture feature extraction and analysis tasks. This point confirms our previous observation concerning the drawback of the *"HBR2013 dataset"* which it does not seem neither very realistic/representative nor appropriate in view of meeting the need to analyze properly texture features (*cf.* Figure 4.13). For instance, an example of a resulting image of the proposed Gabor-based pixel-labeling scheme applied on a HDI from the *"HBR2013 dataset"* illustrating the identification of the vertical black borders and the copyright notice at the bottom of the page as a class (green) on its own and the two different text fonts (blue) together in an another class (*cf.* Figure 4.13(a)). We note that the results would be better if the copyright notices at the bottom of pages under consideration should be either considered and labeled in an another ground-truth class different of the already defined ones or removed from the involved pages. Two examples of the *"Three fonts and graphics"* category of *"HBR2013 dataset"* HDIs are illustrated in Figures 4.18 and 4.19, respectively. The first example of the *"Three fonts and graphics"* category of *"HBR2013 dataset"* HDIs (*cf.* Figure 4.18) shows that the Gabor features give the best results in terms of the homogeneity

of the textual region content (*cf.* Figure 4.18(f)). A cluster representing the uppercase-text font (blue) is clearly identified when analyzing the Gabor features on Figure 4.18(f). However, a slight confusion is also observed between the pixels of the uppercase-text font (blue) and the graphical regions (green) (*cf.* Figure 4.18(f)). The pixel-labeling results by analyzing the Db3 wavelet features (*cf.* Figure 4.18(h)) are similar to those obtained by analyzing the Gabor descriptors when the HDI contains three fonts and graphics (*cf.* Figure 4.18(f)). By visual inspection of the second example of the *"Three fonts and graphics"* category of *"HBR2013 dataset"* HDIs, we observe a slight outperformance of the pixel-labeling results in terms of the homogeneity of the textual region content for the auto-correlation, Gabor and the three investigated wavelets features (*cf.* Figures 4.19(d), 4.19(f), 4.19(g), 4.19(h) and 4.19(i)). The three different text fonts are grouped in one class while the graphical regions are distributed into three classes according to the orientation of the graphical content. This strengthens our previous observations that there is a clear need for first discriminating text from graphic regions and then separating the different text fonts by means of recursive clustering methods.

(a) *"Only two fonts"* (b) *"Only three fonts"*

(c) *"Three fonts and graphics"* (d) *"Four fonts and graphics"*

Figure 4.13.: Examples of resulting images of the proposed Gabor-based pixel-labeling scheme, illustrating few drawbacks of using the *"HBR2013 dataset"* for analyzing texture features.

(a) ***Tamura***
$J = 0.44$ $PPB = 0.90$
$F = 0.57$ $CA = 0.60$

(b) ***LBP***
$J = 0.43$ $PPB = 0.89$
$F = 0.74$ $CA = 0.68$

(c) ***GLRLM***
$J = 0.37$ $PPB = 0.74$
$F = 0.58$ $CA = 0.55$

(d) ***Auto-correlation***
$J = 0.43$ $PPB = 0.92$
$F = 0.61$ $CA = 0.52$

(e) ***GLCM***
$J = 0.49$ $PPB = 0.85$
$F = 0.76$ $CA = 0.77$

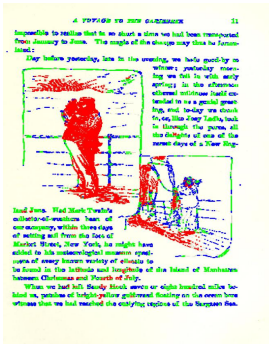
(f) ***Gabor***
$J = 1$ $PPB = 1$
$F = 1$ $CA = 1$

(g) ***Haar***
$J = 0.38$ $PPB = 0.84$
$F = 0.54$ $CA = 0.57$

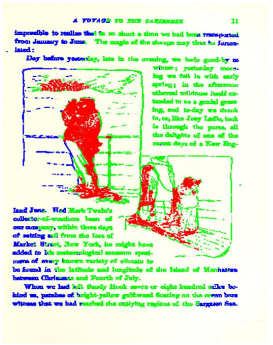
(h) ***Db3***
$J = 0.46$ $PPB = 0.94$
$F = 0.57$ $CA = 0.47$

(i) ***Db4***
$J = 0.46$ $PPB = 0.92$
$F = 0.51$ $CA = 0.45$

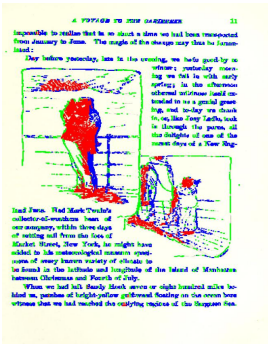
Figure 4.14.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Only two fonts"*** category of HDIs from the ***"HBR2013 dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



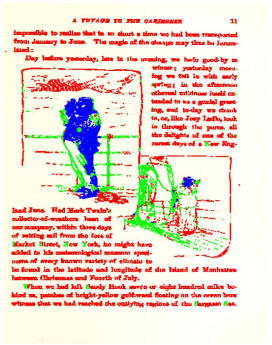
(a) ***Tamura***
$J = 0.41$ $PPB = 0.61$
$F = 0.57$ $CA = 0.62$



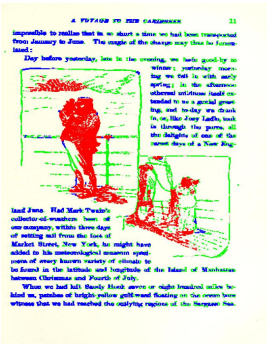
(b) ***LBP***
$J = 0.42$ $PPB = 0.69$
$F = 0.58$ $CA = 0.46$



(c) ***GLRLM***
$J = 0.36$ $PPB = 0.62$
$F = 0.53$ $CA = 0.57$



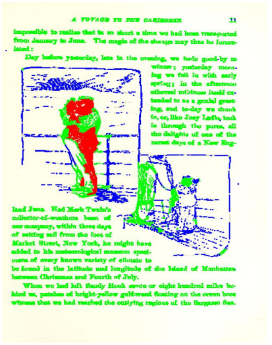
(d) ***Auto-correlation***
$J = 0.50$ $PPB = 0.85$
$F = 0.52$ $CA = 0.68$



(e) ***GLCM***
$J = 0.53$ $PPB = 0.73$
$F = 0.61$ $CA = 0.71$

(f) ***Gabor***
$J = 0.74$ $PPB = 0.97$
$F = 0.58$ $CA = 0.74$

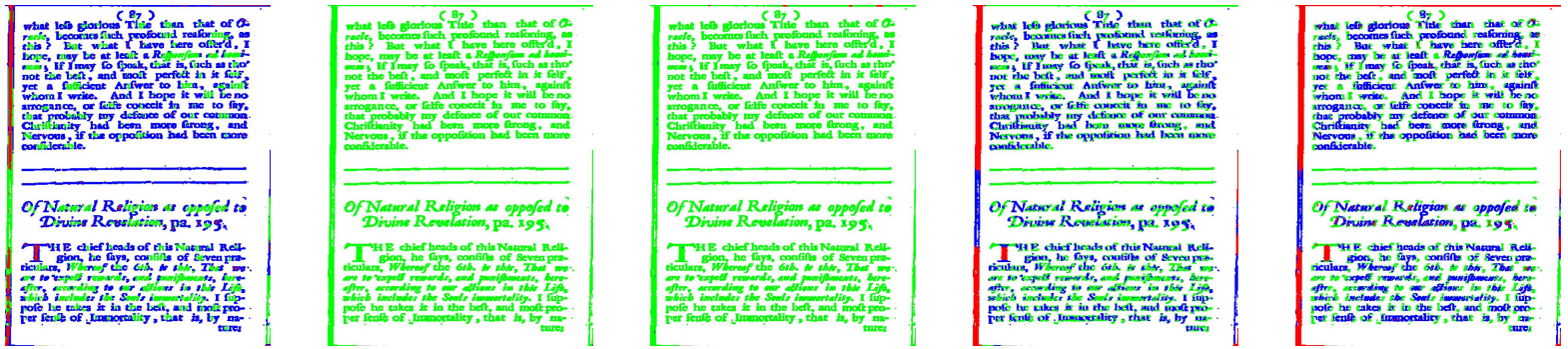
(g) ***Haar***
$J = 0.55$ $PPB = 0.88$
$F = 0.53$ $CA = 0.70$



(h) ***Db3***
$J = 0.52$ $PPB = 0.87$
$F = 0.52$ $CA = 0.70$

(i) ***Db4***
$J = 0.73$ $PPB = 0.95$
$F = 0.59$ $CA = 0.80$

Figure 4.15.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Two fonts and graphics"*** category of HDIs from the ***"HBR2013 dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



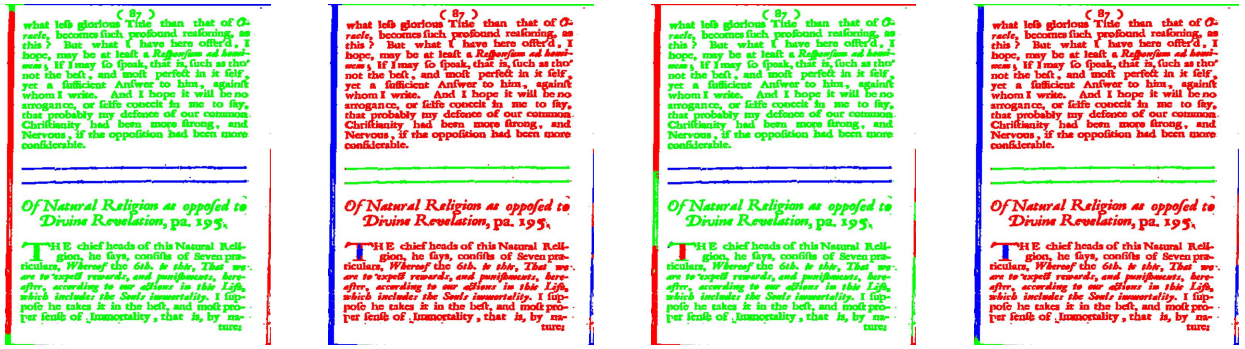
(a) *Tamura*
$J = 0.47$ $PPB = 0.80$
$F = 0.41$ $CA = 0.28$

(b) *LBP*
$J = 0.68$ $PPB = 1$
$F = 0.30$ $CA = 0.81$

(c) *GLRLM*
$J = 0.68$ $PPB = 0.99$
$F = 0.35$ $CA = 0.81$

(d) *Auto-correlation*
$J = 0.42$ $PPB = 0.68$
$F = 0.48$ $CA = 0.53$

(e) *GLCM*
$J = 0.46$ $PPB = 0.73$
$F = 0.52$ $CA = 0.62$



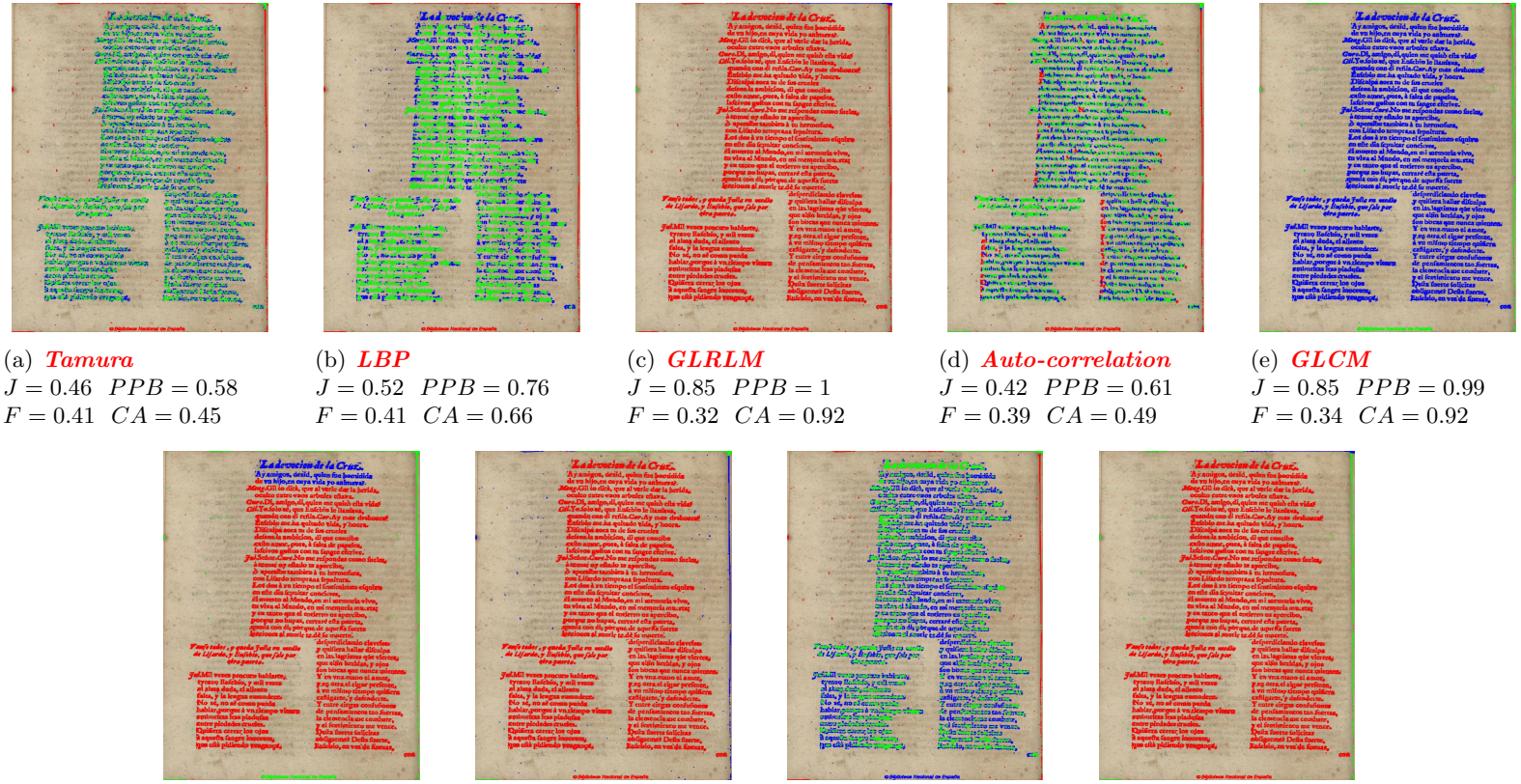
(f) *Gabor*
$J = 0.76$ $PPB = 1$
$F = 0.62$ $CA = 0.87$

(g) *Haar*
$J = 0.77$ $PPB = 0.95$
$F = 0.61$ $CA = 0.87$

(h) *Db3*
$J = 0.77$ $PPB = 0.95$
$F = 0.62$ $CA = 0.87$

(i) *Db4*
$J = 0.77$ $PPB = 0.95$
$F = 0.62$ $CA = 0.87$

Figure 4.16.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Two fonts and graphics"*** category of HDIs from the ***"HBR2013 dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



(a) ***Tamura***
$J = 0.46$ $PPB = 0.58$
$F = 0.41$ $CA = 0.45$

(b) ***LBP***
$J = 0.52$ $PPB = 0.76$
$F = 0.41$ $CA = 0.66$

(c) ***GLRLM***
$J = 0.85$ $PPB = 1$
$F = 0.32$ $CA = 0.92$

(d) ***Auto-correlation***
$J = 0.42$ $PPB = 0.61$
$F = 0.39$ $CA = 0.49$

(e) ***GLCM***
$J = 0.85$ $PPB = 0.99$
$F = 0.34$ $CA = 0.92$

(f) ***Gabor***
$J = 0.83$ $PPB = 0.99$
$F = 0.57$ $CA = 0.91$

(g) ***Haar***
$J = 0.85$ $PPB = 0.99$
$F = 0.34$ $CA = 0.92$

(h) ***Db3***
$J = 0.47$ $PPB = 0.61$
$F = 0.44$ $CA = 0.56$

(i) ***Db4***
$J = 0.85$ $PPB = 0.99$
$F = 0.35$ $CA = 0.92$

Figure 4.17.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Only three fonts"*** category of HDIs from the ***"HBR2013 dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.

(a) ***Tamura***
$J = 0.20$ $PPB = 0.57$
$F = 0.35$ $CA = 0.27$

(b) ***LBP***
$J = 0.39$ $PPB = 0.79$
$F = 0.49$ $CA = 0.56$

(c) ***GLRLM***
$J = 0.41$ $PPB = 0.78$
$F = 0.48$ $CA = 0.63$

(d) ***Auto-correlation***
$J = 0.42$ $PPB = 0.66$
$F = 0.55$ $CA = 0.64$

(e) ***GLCM***
$J = 0.52$ $PPB = 0.80$
$F = 0.58$ $CA = 0.52$

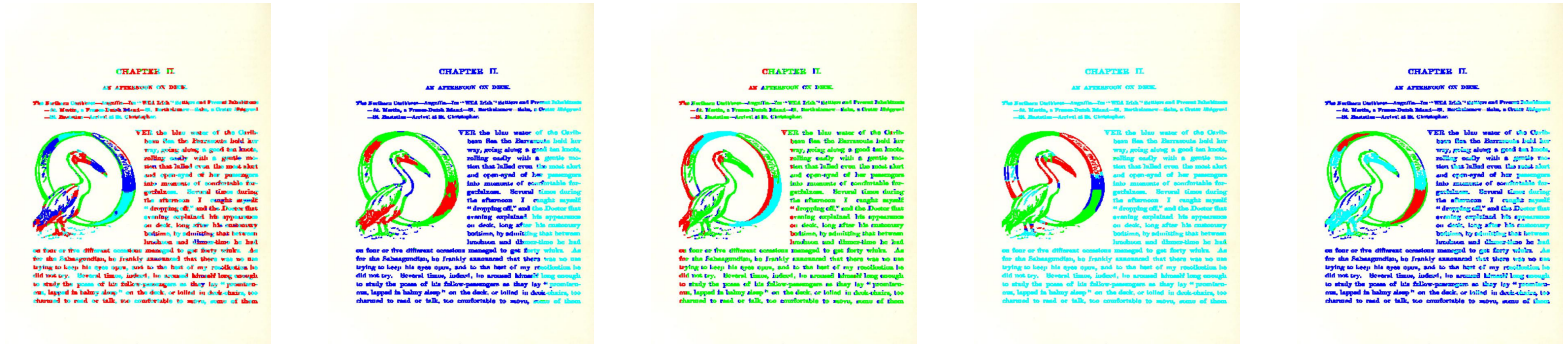
(f) ***Gabor***
$J = 0.65$ $PPB = 0.97$
$F = 0.71$ $CA = 0.60$

(g) ***Haar***
$J = 0.46$ $PPB = 0.73$
$F = 0.56$ $CA = 0.69$

(h) ***Db3***
$J = 0.45$ $PPB = 0.86$
$F = 0.48$ $CA = 0.56$

(i) ***Db4***
$J = 0.49$ $PPB = 0.81$
$F = 0.61$ $CA = 0.64$

Figure 4.18.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Three fonts and graphics"*** category of HDIs from the ***"HBR2013 dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



(a) ***Tamura***
$J = 0.31$ $PPB = 0.58$
$F = 0.40$ $CA = 0.45$

(b) ***LBP***
$J = 0.42$ $PPB = 0.79$
$F = 0.48$ $CA = 0.39$

(c) ***GLRLM***
$J = 0.37$ $PPB = 0.72$
$F = 0.40$ $CA = 0.55$

(d) ***Auto-correlation***
$J = 0.59$ $PPB = 0.95$
$F = 0.42$ $CA = 0.74$

(e) ***GLCM***
$J = 0.36$ $PPB = 0.80$
$F = 0.42$ $CA = 0.41$

(f) ***Gabor***
$J = 0.64$ $PPB = 0.96$
$F = 0.44$ $CA = 0.77$

(g) ***Haar***
$J = 0.58$ $PPB = 0.93$
$F = 0.40$ $CA = 0.66$

(h) ***Db3***
$J = 0.58$ $PPB = 0.92$
$F = 0.39$ $CA = 0.66$

(i) ***Db4***
$J = 0.63$ $PPB = 0.96$
$F = 0.43$ $CA = 0.76$

Figure 4.19.: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the ***"Three fonts and graphics"*** category of HDIs from the ***"HBR2013 dataset"***. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.

4.5.1.3. Performance evaluation

This way of comparing visually the effectiveness of a texture-based approach is inherently a subjective evaluation. Therefore, finding appropriate quantitative accuracy metrics is required first to evaluate the performance of the obtained results of the proposed pixel-labeling scheme for comparing the nine investigated texture feature sets. Then, it is necessary to assess quantitatively the results of the proposed pixel-labeling scheme for comparing the nine investigated texture feature sets in order to have a conclusion of which texture methods are firstly well suited for segmenting graphical regions from textual ones and discriminating text in a variety of situations of different fonts and scales ? As a consequence, in this work several clustering and classification accuracy measures are computed, silhouette width index (SW), Jaccard coefficient (J), purity per block (PPB), F-measure (F) and classification accuracy rate (CA), to evaluate quantitatively the obtained results of the proposed pixel-labeling scheme for comparing the nine investigated texture feature sets (*cf.* Section 4.4.3). The higher the values, the better the results. We will limit ourselves here therefore to calculate the SW metric only for the *"DIGIDOC-Texture dataset"* because its computation is very time-consuming process.

In Tables 4.2, 4.6, 4.7 and 4.8, there are two *"Overall"* values. The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. The *"Two fonts and graphics*"* value represents the case when every font in the text has a distinct label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 3 (graphics and text with two different fonts). The *"Two fonts and graphics**"* value represents the case when all fonts in the text have the same label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text). This distribution points out which texture features can be more adequate for segmenting documents containing two text fonts and graphics into two/three classes, *i.e.* separating two distinct text fonts when the documents contain graphics.

The comparison results produced by using the nine texture-based feature sets in the proposed pixel-labeling scheme on the two datasets, the DIGIDOC-Texture and HBR2013 datasets, are presented in Tables 4.2 and 4.3, respectively.

1. ***"DIGIDOC-Texture dataset"***
   In Table 4.2, the computed clustering and classification accuracy values are congruent and very promising. However, we note a slight difference in the performance of the SW average and slight variability in the ranking of the different investigated texture-based feature sets when computing the SW metric. This can be explained by the progressive merge process of the HAC algorithm used in the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets, where in higher levels in the hierarchy, two distant data points can be merged together and yet still belong to the same cluster after cutting the dendrogram. This causes a slightly lower value of the SW. This justification can be strengthened by the particularity of the SW as internal or unsupervised accuracy clustering evaluation which investigates the coherence of a clustering solution by measuring how observations are close to the cluster center and how clusters are well-separated.

   We observe that the best average performances for most of the computed evaluation metrics are obtained by the Gabor features (75%(J), 93%(PPB), 76%(F) and 78%(CA) for *"Overall*"*, and 80%(J), 94%(PPB), 81%(F) and 82%(CA) for *"Overall**"*). When using *"Two fonts and graphics**"* in computing *"Overall**"*, we observe that the performance of the extracted Gabor descriptors is much better when using *"Two fonts and graphics*"* in computing *"Overall*"*, *i.e.* overall performance gains of 5%(J), 1%(PPB), 5%(F) and 4%(CA) are noted. This strengthens our previous observations in Section 4.5.1.2 that there is a clear need for first discriminating text from graphic regions and then separating the different text fonts by means of recursive clustering methods to have better performance. Here, it can be

observed that the result of employing the Gabor features yields a better output than the eight other extracted textural features for almost all evaluation accuracy metrics without taking into consideration the spatial relationships of pixels. We note that the best results of mean F values are obtained by the Gabor features for almost HDI categories of the *"DIGIDOC-Texture dataset"* (88%, 67%, 89%, 84% and 64% for the *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* HDI categories, respectively). Similarly, the best results of mean PPB values are observed when analyzing the Gabor features (96%, 93%, 98%, 94% and 88% for the *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* categories, respectively). This can be explained by the optimal localization properties of GFs to capture information in both the spatial and frequency domains from the analyzed HDIs (*i.e.* GFs are inherently multi-resolutional).

We note that the second best performance is obtained for almost all HDI categories of the *"DIGIDOC-Texture dataset"* when using one of three investigated kinds of wavelet features on the proposed pixel-labeling scheme. This can be justified by the consistent properties of the wavelet features in the localization of the frequency space and multi-resolution. When using the Db4 features, 84%, 63%, 89%, 76% and 59% of F values are noted for the *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* categories, respectively. Low values of performance difference of the computed evaluation metrics between the used Gabor and wavelet features on the proposed pixel-labeling scheme when HDIs containing graphics and text (F difference values of 4%, 4% and 0% for the *"One font and graphics"*, *"Two fonts and graphics*"* and *"Two fonts and graphics**"* HDI categories, respectively) compared to the case when HDIs containing only text (F difference values of 8% and 5% for the *"Only two fonts"* and *"Only three fonts"* HDI categories, respectively). We conclude that the Gabor-based approach performs considerably better than the wavelet one if the analyzed HDI contains only text. Nevertheless, the values of the computed accuracy metrics are low with the *"Only three fonts"* category (0.31(SW), 60%(J), 88%(PPB), 64%(F) and 68%(CA) are noted when using the Gabor-based approach) comparing with the *"One font and graphics"* (difference values of 0.21(SW), 28%(J), 8%(PPB), 24%(F) and 19%(CA)). As a consequence, the Gabor-based approach performs significantly better than the other investigated features specifically when the involved HDI contains two different text fonts or graphics and text. This strengthens our previous observations obtained when analyzing visually the results and confirms our assumption that the Gabor descriptors are the most suitable for font segmentation, since they are known to be sensitive to the stroke width.

We also observe that the performance values of the computed accuracy metrics for almost all HDI categories of the *"DIGIDOC-Texture dataset"* when using the auto-correlation descriptors are close to those when using the Gabor and wavelet features (82%, 59%, 83%, 72% and 61% of F values are noted for the *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* HDI categories, respectively). This can be justified that the auto-correlation attributes mainly provide an interesting information about the main orientation of a texture which can be relevant for discriminating between the different classes of the foreground layers.

Overall, the worst performances are mainly obtained when using the GLRLM features on the proposed pixel-labeling scheme. We have found that the GLRLM features gave the worst overall performances for most of the computed evaluation metrics (60%(J), 82%(PPB) and 53%(F) for *"Overall*"* category, and 67%(J), 85%(PPB) and 57%(F) for *"Overall**"*). This can be justified by the fact that these features are so simple and they can not provide adequate information needs to characterize a texture, neither an information concerning the main orientation of a texture for discriminating graphical regions from the textual ones, nor an indice about the stroke properties for text font segmentation. For HDIs containing only

distinct fonts, we observe that the lowest values of the computed clustering and classification accuracy metrics are divided among multiple texture-based feature sets (e.g. Tamura, GLRLM and GLCM descriptors). Therefore, we conclude that the Tamura, GLRLM and GLCM features are not adequate for separating different text fonts even when they are the lowest time-consuming.

2. *"**HBR2013 dataset**"*
In Table 4.3, we observe that calculating the overall accuracy metrics on the *"HBR2013 dataset"* confirms the results obtained by using the *"DIGIDOC-Texture dataset"*. The Gabor-based approach is the best one (overall values of 53%(J), 91%(PPB), 51%(F) and 59%(CA) are noted). However, we note a significant drop in performance of the 22%(J), 2%(PPB), 25%(F) and 19%(CA) when applying the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"HBR2013 dataset"* comparing the *"DIGIDOC-Texture dataset"*. This can be justified by the produced bias in the texture feature extraction and analysis tasks due to the drawbacks of the *"HBR2013 dataset"* which it does not seem neither very realistic/representative nor appropriate in view of meeting the need to analyze properly texture features (e.g. binary HDIs, low resolution digitization, presence of copyright notices in many pages). Moreover, the *"HBR2013 dataset"* is complex since the values of the number of types of content regions defined in the ground-truth are distributed across the $[2, 6]$ range. Unlike the *"HBR2013 dataset"*, the values of the number of types of content regions defined in the ground-truth of the *"DIGIDOC-Texture dataset"* is equal to either 2 or 3. In addition, as we mentioned before that there is a clear need for first discriminating text from graphic regions and then separating the different text fonts by means of recursive clustering methods to have better performance. Thus, the performance of the results depends on the values of the number of types of content regions defined in the ground-truth. The smaller values of the number of types of content regions defined in the ground-truth represent higher efficiency. We note that the performance decreases since the number of text fonts increases (75%, 52%, 44% and 41% of J for the *"Only two fonts"*, *"Only three fonts"*, *"Only four fonts"* and *"Only five fonts"* HDIs categories).

We note that the second best performance is obtained for almost all HDI categories of the *"HBR2013 dataset"* when using one of three investigated kinds of wavelet features on the proposed pixel-labeling scheme (74%, 52%, 39% and 41% of J for the *"Only two fonts"*, *"Only three fonts"*, *"Only four fonts"* and *"Only five fonts"* HDIs categories). The results obtained with the *"HBR2013 dataset"* strengthen our previous observations with the *"DIGIDOC-Texture dataset"*.

We observe that the worst performances are mainly obtained when using the Tamura and GLRLM features on the proposed pixel-labeling scheme. When using the Tamura features, 90%, 81%, 78% and 70% of PPB values are noted for the *"Only two fonts"*, *"Two fonts and graphics"*, *"Only three fonts"* and *"Three fonts and graphics"* categories of the *"DIGIDOC-Texture dataset"*, respectively. On the other side, when using the GLRLM features, 50%, 42%, 37%, 37%, 29%, 29%, 22% and 29% of F values are noted for the *"Only two fonts"*, *"Two fonts and graphics"*, *"Only three fonts"*, *"Three fonts and graphics"*, *"Only four fonts"*, *"Four fonts and graphics"*, *"Only five fonts"*, *"Five fonts and graphics"* categories of the *"DIGIDOC-Texture dataset"*, respectively. The worst overall performances for most of the computed evaluation metrics are noted when using the Tamura features on the proposed pixel-labeling scheme (47%(J), 71%(PPB) and 39%(CA)).

Table 4.2.: Evaluation of the analyzed textural features on the *"DIGIDOC-Texture dataset"*. Clustering and classification accuracy measures are computed: silhouette width (SW), Jaccard coefficient (J), purity per block (PPB), F-measure (F) and classification accuracy (CA). The higher the values, the better the results. The values which are quoted in **red** and **green** colors, are considered as the **lowest** and **highest**, respectively.

		Tamura	**LBP**	**GLRLM**	**Auto-correlation**	**GLCM**	**Gabor**	**Haar**	**Db3**	**Db4**
One font and graphics	SW	**0.39**	0.57	**0.70**	0.54	0.46	0.51	0.56	0.58	0.57
	J	0.73	0.73	**0.70**	0.79	0.79	**0.88**	0.84	0.85	0.87
	PPB	**0.90**	0.91	0.92	0.91	0.92	**0.96**	0.95	0.95	0.95
	F	0.74	0.74	**0.64**	0.82	0.78	**0.88**	0.82	0.84	0.84
	CA	0.69	0.73	**0.66**	0.85	0.71	**0.87**	0.74	0.78	0.77
Two fonts and graphics*	SW	**0.16**	0.37	0.28	0.27	0.30	**0.43**	0.38	0.41	0.40
	J	0.59	0.58	**0.55**	0.61	0.60	**0.70**	0.65	0.65	0.67
	PPB	0.86	0.83	**0.80**	0.83	0.86	**0.93**	0.89	0.88	0.91
	F	0.55	0.60	**0.52**	0.59	0.61	**0.67**	0.63	0.62	0.63
	CA	**0.63**	0.66	0.64	0.70	0.68	0.75	0.73	0.74	**0.76**
Two fonts and graphics**	SW	**0.27**	**0.59**	0.54	0.52	0.47	0.48	0.53	0.56	0.54
	J	0.90	0.93	**0.85**	0.89	0.94	0.91	0.91	**0.95**	0.92
	PPB	0.93	0.93	**0.89**	0.90	0.94	**0.98**	0.96	0.96	0.97
	F	0.79	0.81	**0.69**	0.83	0.84	**0.89**	0.87	**0.89**	**0.89**
	CA	0.81	0.82	**0.74**	0.84	0.85	0.89	0.87	**0.90**	**0.90**
Only two fonts	SW	**0.23**	0.37	**0.66**	0.27	0.25	0.39	0.30	0.31	0.30
	J	0.65	0.62	0.66	0.71	**0.60**	**0.82**	0.69	0.65	0.70
	PPB	0.87	0.85	0.87	0.84	**0.82**	**0.94**	0.88	0.85	0.88
	F	0.59	0.69	**0.55**	0.72	0.70	**0.84**	0.74	0.73	0.76
	CA	**0.52**	0.67	0.61	0.78	0.67	**0.82**	0.76	0.75	0.78

Table 4.2 – continued from previous page

		Tamura	LBP	GLRLM	Auto-correlation	GLCM	Gabor	Haar	Db3	Db4
Only three fonts	*SW*	0.16	0.19	0.14	**0.07**	0.15	**0.31**	0.18	0.22	0.19
	J	0.51	**0.46**	0.47	**0.61**	**0.46**	0.60	0.50	0.48	0.52
	PPB	0.84	0.74	**0.70**	0.77	0.74	**0.88**	0.78	0.76	0.79
	F	0.43	0.54	**0.41**	0.61	0.60	**0.64**	0.56	0.57	0.59
	CA	**0.41**	0.57	0.47	0.62	0.65	**0.68**	0.61	0.61	0.64
Overall*	***SW***	**0.24**	**0.38**	**0.45**	**0.29**	**0.29**	**0.41**	**0.36**	**0.38**	**0.37**
	J	**0.62**	**0.60**	**0.60**	**0.68**	**0.61**	**0.75**	**0.67**	**0.66**	**0.69**
	PPB	**0.87**	**0.83**	**0.82**	**0.84**	**0.84**	**0.93**	**0.88**	**0.86**	**0.88**
	F	**0.58**	**0.64**	**0.53**	**0.69**	**0.67**	**0.76**	**0.69**	**0.69**	**0.71**
	CA	**0.56**	**0.66**	**0.60**	**0.74**	**0.68**	**0.78**	**0.71**	**0.72**	**0.74**
Overall**	***SW***	**0.26**	**0.43**	**0.51**	**0.35**	**0.33**	**0.42**	**0.39**	**0.42**	**0.40**
	J	**0.70**	**0.69**	**0.67**	**0.75**	**0.70**	**0.80**	**0.74**	**0.73**	**0.75**
	PPB	**0.89**	**0.86**	**0.85**	**0.86**	**0.86**	**0.94**	**0.89**	**0.88**	**0.90**
	F	**0.64**	**0.70**	**0.57**	**0.75**	**0.73**	**0.81**	**0.75**	**0.76**	**0.77**
	CA	**0.61**	**0.70**	**0.62**	**0.77**	**0.72**	**0.82**	**0.75**	**0.76**	**0.77**

Table 4.3.: Evaluation of the analyzed textural features on the *"HBR2013 dataset"*. Clustering and classification accuracy measures are computed: Jaccard coefficient (J), purity per block (PPB), F-measure (F) and classification accuracy (CA). The higher the values, the better the results. The values which are quoted in **red** and **green** colors, are considered as the **lowest** and **highest**, respectively.

		Tamura	**LBP**	**GLRLM**	**Auto-correlation**	**GLCM**	**Gabor**	**Haar**	**Db3**	**Db4**
Only two fonts	J	0.75	**0.71**	**0.83**	0.80	0.74	0.75	0.73	**0.71**	0.74
	PPB	**0.90**	0.92	**0.96**	0.95	0.92	0.94	0.91	0.92	0.92
	F	0.54	0.55	**0.50**	0.53	0.56	**0.59**	0.56	0.57	0.56
	CA	0.73	**0.62**	0.70	0.65	0.63	0.64	0.68	**0.75**	0.71
Two fonts and graphics	J	0.60	0.68	**0.75**	0.66	**0.55**	0.63	0.68	0.61	0.71
	PPB	**0.81**	0.91	0.91	0.84	0.82	**0.92**	0.88	0.88	0.90
	F	0.50	0.43	**0.42**	0.53	0.54	0.55	0.60	0.58	**0.62**
	CA	**0.46**	0.66	0.49	0.71	0.56	0.69	**0.79**	0.71	0.78
Only three fonts	J	0.53	0.53	**0.65**	0.58	0.59	0.52	0.52	**0.51**	0.52
	PPB	**0.78**	0.83	**0.87**	0.80	**0.87**	**0.87**	0.83	0.83	0.83
	F	0.39	0.41	**0.37**	0.42	0.40	**0.53**	0.43	0.42	0.43
	CA	**0.34**	0.46	**0.60**	0.47	0.57	0.48	0.44	0.46	0.45
Three fonts and graphics	J	0.42	0.43	**0.41**	0.49	0.46	**0.58**	0.49	0.53	0.54
	PPB	**0.70**	0.81	0.80	0.82	0.84	**0.95**	0.86	0.89	0.91
	F	**0.37**	0.40	**0.37**	0.41	0.44	**0.53**	0.41	0.41	0.42
	CA	**0.33**	0.48	0.52	0.61	0.50	0.64	0.59	0.62	**0.65**
Only four fonts	J	**0.48**	0.41	0.46	0.46	0.44	0.44	**0.39**	0.40	**0.39**
	PPB	0.77	0.76	0.76	**0.72**	0.85	**0.90**	0.73	0.74	0.75
	F	0.32	0.32	**0.29**	0.41	0.34	**0.43**	0.37	0.39	0.37
	CA	**0.37**	0.42	0.37	0.45	0.43	**0.55**	0.47	0.46	0.40

Table 4.3 – continued from previous page

		Tamura	LBP	GLRLM	Auto-correlation	GLCM	Gabor	Haar	Db3	Db4
Four fonts and graphics	J	0.38	**0.36**	0.41	0.37	0.43	**0.46**	0.40	0.38	0.40
	PPB	**0.64**	0.77	0.76	0.67	0.81	**0.90**	0.79	0.78	0.79
	F	0.33	**0.29**	**0.29**	0.37	0.38	**0.44**	0.41	0.41	0.43
	CA	**0.31**	0.42	0.40	0.45	0.45	**0.58**	0.50	0.48	0.51
Only five fonts	J	**0.28**	0.36	0.40	0.33	**0.44**	0.41	0.41	0.34	0.41
	PPB	**0.52**	0.69	0.77	0.59	0.86	**0.89**	0.85	0.73	0.87
	F	0.30	0.24	**0.22**	0.30	0.31	**0.45**	0.29	0.32	0.30
	CA	**0.28**	0.39	0.49	0.31	0.37	**0.58**	0.46	0.42	0.45
Five fonts and graphics	J	0.29	0.29	**0.26**	0.31	0.40	**0.42**	0.33	0.36	0.39
	PPB	0.59	0.64	**0.56**	0.61	0.75	**0.88**	0.68	0.66	0.70
	F	**0.29**	0.31	**0.29**	0.39	0.39	**0.55**	0.41	0.44	0.44
	CA	**0.31**	0.39	0.33	0.35	0.46	**0.55**	0.44	0.44	0.50
Overall	***J***	**0.47**	**0.47**	**0.52**	**0.50**	**0.51**	**0.53**	**0.49**	**0.48**	**0.51**
	PPB	**0.71**	**0.79**	**0.80**	**0.75**	**0.84**	**0.91**	**0.82**	**0.80**	**0.83**
	F	**0.38**	**0.37**	**0.34**	**0.42**	**0.42**	**0.51**	**0.43**	**0.44**	**0.45**
	CA	**0.39**	**0.48**	**0.49**	**0.50**	**0.50**	**0.59**	**0.54**	**0.54**	**0.56**

4.5.1.4. Recommendations

Based on the experimental results and observations (*cf.* Section 4.5.1, few recommendations have been deduced about the choice of the used texture feature set. These recommendations are based on analyzing texture features without formulating a hypothesis concerning the HDI layout (e.g. column layout) or its content (e.g. font size and type) and respecting a constructive compromise between the pixel-labeling quality, performance evaluation (*cf.* Sections 4.5.1.2 and 4.5.1.3) and computational cost (*cf.* Table 4.5).

- The performances of the Tamura, LBP and GLRLM-based approaches are less satisfactory particularly for HDIs containing only text, compared to the other investigated texture-based approaches even the numerical complexity is sufficiently adequate.
- The GLCM-based approach should be a good choice for HDIs containing graphics and single text font as it is fast and easy to use. Indeed, the lowest time required to process a page is obtained when using the GLCM descriptors. Nevertheless, the GLCM features are not adequate for separating different text fonts even when it is the lowest time-consuming.
- The auto-correlation approach is an effective and efficient texture-based one, particularly for HDIs containing graphics and text.
- The auto-correlation and GLCM features perform considerably better when the HDI under consideration containing graphics and text than only text.
- The computational cost of using the auto-correlation and LBP features is similar. However, the auto-correlation-based approach performs considerably better than the LBP one when comparing their pixel-labeling quality and computed accuracy metrics.
- The wavelet-based approach is more suitable for distinguishing textual regions from graphical ones. However, when the numerical complexity is taken into account, the wavelet-based approach is the highest resource-consuming one.
- The two kinds of wavelet features, Db3 and Db4, perform better than the Haar ones for all kinds of HDI content. The counterpart for the robustness of using the Db4 and Db3 features is a higher computing time.
- The Gabor-based approach performs considerably better in segmenting HDIs containing only textual regions with distinct fonts.
- The best performing kind of texture features is the Gabor ones for all types of HDI content. The Gabor-based approach yields a better output than the eight other extracted textural features for almost all evaluation accuracy metrics without taking into consideration the spatial relationships of pixels. Nevertheless, the feature dimension of the Gabor-based approach is relatively high. This requires a relatively higher computing time and a lot of computer memory (*i.e.* quite resource-consuming).
- When the numerical complexity and performance evaluation are taken into account by comparing the two best investigated texture-based approaches (*i.e.* the Gabor and wavelet-based approaches), the Gabor one would be the better choice for segmenting graphical regions from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand, without formulating a hypothesis concerning the HDI layout (e.g. column layout) or its content (e.g. font size and type).

Table 4.4.: Computational cost of the texture feature analysis task (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality): an example of HDI (1965×2750 pixels). The values which are quoted in **red** and **green** colors, are considered as the **lowest** and **highest**, respectively.

	Tamura	**LBP**	**GLRLM**	**Auto-correlation**	**GLCM**	**Gabor**	**Haar**	**Db3**	**Db4**
Running time	$01'14''$	$02'24''$	$00'32''$	$02'33''$	$00'14''$	$06'05''$	$29'17''$	$37'53''$	$42'21''$
Used memory	≈94 MB	≈53 MB	≈82 MB	**≈48 MB**	**≈587 MB**	≈552 MB	≈61 MB	≈61 MB	≈63 MB
Complexity	$O(Mn_t 2^{2k})$	$O(MP2^P)$	$O(M\theta_r n_r)$	$O(M(\theta_a N_w \log_2 N_w))$	$O(Md_c n_g^2)$	$O(f_g\theta_g(S^2 \log_2 S))$	$O(M(4JN_w^2 \log_2 N_w))$	$O(M(6JN_w^2 \log_2 N_w))$	$O(M(8JN_w^2 \log_2 N_w))$
Texture vector size	$16 = I_t \times N_w$	$40 = I_l \times N_w$	$176 = I_r \times N_w$	$20 = I_a \times N_w$	$72 = I_c \times N_w$	$192 = I_g \times N_w$	$80 = I_h \times N_w$	$80 = I_{db3} \times N_w$	$80 = I_{db4} \times N_w$
Number of the texture indices	$I_t = 4$	$I_l = 10$	$I_r = 11\theta_r = 44$	$I_a = 5$	$I_c = 8d_c + 2 = 18$	$I_g = 2f_g\theta_g = 48$	$I_h = 2I_{A_{2^{-J}}} + 2I_{D^{(v)}_{2^{-1}}} + \cdots + 2I_{D^{(v)}_{2^{-j}}} + \cdots + 2I_{D^{(v)}_{2^{-J}}} + 2I_{D^{(h)}_{2^{-1}}} + \cdots + 2I_{D^{(h)}_{2^{-j}}} + \cdots + 2I_{D^{(h)}_{2^{-J}}} + 2I_{D^{(d)}_{2^{-1}}} + \cdots + 2I_{D^{(d)}_{2^{-j}}} + \cdots + 2I_{D^{(d)}_{2^{-J}}} = 20$	$I_{db3} = 2I_{A_{2^{-J}}} + 2I_{D^{(v)}_{2^{-1}}} + \cdots + 2I_{D^{(v)}_{2^{-j}}} + \cdots + 2I_{D^{(v)}_{2^{-J}}} + 2I_{D^{(h)}_{2^{-1}}} + \cdots + 2I_{D^{(h)}_{2^{-j}}} + \cdots + 2I_{D^{(h)}_{2^{-J}}} + 2I_{D^{(d)}_{2^{-1}}} + \cdots + 2I_{D^{(d)}_{2^{-j}}} + \cdots + 2I_{D^{(d)}_{2^{-J}}} = 20$	$I_{db4} = 2I_{A_{2^{-J}}} + 2I_{D^{(v)}_{2^{-1}}} + \cdots + 2I_{D^{(v)}_{2^{-j}}} + \cdots + 2I_{D^{(v)}_{2^{-J}}} + 2I_{D^{(h)}_{2^{-1}}} + \cdots + 2I_{D^{(h)}_{2^{-j}}} + \cdots + 2I_{D^{(h)}_{2^{-J}}} + 2I_{D^{(d)}_{2^{-1}}} + \cdots + 2I_{D^{(d)}_{2^{-j}}} + \cdots + 2I_{D^{(d)}_{2^{-J}}} = 20$

Table 4.5.: Performance evaluation and benchmarking issues of nine investigated texture-based feature sets in this work for segmenting HDIs. The case contents which are quoted in **red** and **green** colors, respectively, are considered as the **worst** and **best**, respectively.

	Tamura	**LBP**	**GLRLM**	**Auto-correlation**	**GLCM**	**Gabor**	**Haar**	**Db3**	**Db4**
Dimensionality	++	++	–	++	+	–	+	+	+
Complexity	+	+	++	+	++	+	–	– –	– –
Used memory	+	+	+	++	–	–	+	+	+
Performance									
One font and graphics	–	–	–	+	+	++	+	++	++
Two fonts and graphics*	–	–	–	+	+	++	+	+	+
Two fonts and graphics**	–	–	–	+	+	++	+	++	++
Only two fonts	– –	– –	– –	–	–	+	–	–	–
Only three fonts	– –	– –	– –	–	–	+	–	–	–

In Table 4.4, I_t, I_l, I_r, I_a, I_c, I_g, I_h, I_{db3} and I_{db4} denote number of extracted Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor, Haar, Db3 and Db4 features, respectively. $I_{A_{2^{-J}}}$, $I_{D^{(v)}_{2^{-j}}}$, $I_{D^{(h)}_{2^{-j}}}$ and $I_{D^{(d)}_{2^{-j}}}$ denote the number of extracted approximation and detail sub-images features. N_w is the number of sliding windows. In this work, N_w is equal to 4. M is the number of foreground pixels. $S = W \times H$ is the dimension or size of the input image. W and H denote the effective width and height of the analyzed image. n_g is the number of gray-levels, *i.e.* 255 gray-levels. n_t is the number of averages $A_{k_t}(x, y)$ for the windows of size $2^{k_t} \times 2^{k_t}$, *i.e.* 3 averages computed around each selected pixel for the windows of size $2^{k_t} \times 2^{k_t}$, where $k_t = \{0, 1, 2\}$. P is the number of LBP neighboring pixels, *i.e.* 8 pixels in the neighbor set. θ_r is the number of angle direction values specified when computing the GLRLM. In this work, θ_r is equal to 4 directions of angle (*i.e.* $\theta_r = \{0°, 45°, 90°, 135°\}$). n_r is the number of pixels of the sliding window. θ_a is the possible number of orientation values of the rose of directions (*i.e.* 179 orientation values). d_c is the GLCM particular distance defined in the probability of the gray-level pairs. In this work, d_c is equal to 2. f_g and θ_g are spatial frequency and orientation of Gabor filters, respectively. In the experiment, the scale of wavelet decomposition J is 3 levels (*i.e.* from first, second and third scale). "MB" means megabytes.

4.5.2. HAC vs. k-means is used in the pixel-clustering task

Another set of experiments has been performed by using two different algorithms, k-means and HAC, in the pixel-clustering task of the pixel-labeling scheme for comparing texture features (*cf.* Figure 4.1, Section 4.4.1.3) in order to compare their performance and to determine which clustering algorithm is more appropriate. To evaluate the two different algorithms, k-means and HAC, we have deliberately limited this comparative study to only the results of few texture features. As we previously have proved that the auto-correlation and Gabor-based approaches are the two effective and efficient texture-based ones (*cf.* Section 4.5.1.4), the results of the auto-correlation and Gabor-based approaches have been compared, using the two different algorithms, k-means and HAC, in this section.

4.5.2.1. Qualitative results

Figures 4.20 and 4.21 illustrate the different resulting HDIs of the pixel-labeling of the extracted auto-correlation and Gabor features using the HAC and k-means algorithms applied on examples of the *"one font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* categories. Measures of F are presented at the bottom of each image in Figures 4.20 and 4.21.

When analyzing visually the overall results, our previous observations concerning the outperformance of the Gabor features comparing the auto-correlation ones have been strengthened even if we change the clustering algorithm. Nevertheless, we observe that the pixel-labeling results for the extracted auto-correlation and Gabor features obtained with the HAC algorithm are quite different from those with the k-means one. The results of the pixel-labeling of the extracted texture features using the HAC and k-means algorithms vary depending on the used texture descriptors and the content and/or layout of the analyzed HDIs. For example, when using the Gabor features, the pixel-labeling results with both the HAC and k-means algorithms are similar in the cases of the analyzed HDI containing one font and graphics, two fonts and graphics where all fonts in the text have the same label in the ground-truth, and only two fonts. However, when using the Gabor features, the pixel-labeling results with the HAC algorithm are better than the k-means one in the case of the analyzed HDI containing only three fonts. The pixel-labeling results for the extracted Gabor features obtained with the HAC algorithm show a much greater discriminating power for discriminating three different text fonts (*cf.* Figure 4.21(f)). Nevertheless, we note that the k-means algorithm performs better than the HAC when using the Gabor features in the case of the

analyzed HDI containing two fonts and graphics where every font in the text has a different label in the ground-truth. As a matter of fact, the pixel-labeling results for the extracted Gabor features obtained with the k-means algorithm show a much greater discriminating power for separating two distinct fonts when the HDI under consideration contains graphics and two different text fonts (*cf.* Figure 4.20(h)).

On the other side, when using the auto-correlation features, the pixel-labeling results with the HAC algorithm performs better than with the k-means one in the cases of the analyzed HDI containing one font and graphics, two fonts and graphics where every font in the text has a different label in the ground-truth, and only two fonts. However, the k-means algorithm shows better results than the HAC when using the auto-correlation features in the cases of the analyzed HDI containing two fonts and graphics where all fonts in the text have the same label in the ground-truth, and only three fonts.

We conclude that there is a variability in the visual results when using the HAC and k-means algorithms in the pixel-clustering task of the pixel-labeling scheme for comparing texture features. This variability in the obtained visual results can be explained by the specificity of each clustering algorithm (e.g. distance as a metric of cluster scatter, initial partitions). Moreover, it can be justified by the dynamic range of the extracted texture attributes which can differ depending on the content and/or layout of the HDI under consideration.

4.5.2.2. Analysis of some examples

In Appendix B and particularly in Section B.2, the confusion matrices and pixel-clustering results given by the HAC and k-means algorithms in the pixel-clustering task of the proposed Gabor-based pixel-labeling scheme on the *"DIGIDOC-Texture dataset"* are illustrated. Figures B.23, B.24 and B.25 illustrate the confusion matrices and resulting HDIs of the pixel-labeling of the extracted Gabor features using the HAC and k-means algorithms applied on examples of the *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* categories of the *"DIGIDOC-Texture dataset"*.

Two examples of confusion matrix computation and pixel-labeling results obtained using the HAC algorithm with the Gabor descriptors are shown in Figures 4.22 and 4.23. Figures 4.22 and 4.23 illustrate the confusion matrices and resulting HDIs of the pixel-labeling of the extracted Gabor features using the HAC and k-means algorithms applied on examples of the *"one font and graphics"* and *"Two fonts and graphics*"*. The elements of the confusion matrix represent the foreground pixels.

– The first example (*cf.* Figure 4.22) presents two confusion matrices and pixel-clustering results with the Gabor descriptors using the HAC and k-means algorithms of a document containing graphics and single text font. The first confusion matrix and pixel-clustering results using the HAC algorithm of a document containing graphics (blue) and single text font (green) is represented on the left. Precision is considered to be a means of assessing the classification while recall is considered as a way of improving the classification. In this example, the graphical pixels are classified with 88%(P) and 98%(R), while for textual ones we find 94%(P) and 71%(R). Hence, the graphic class has a lower precision and a higher recall. Thus, we show that the extracted Gabor descriptors with the HAC algorithm tend to miss more textual pixels than graphical ones (71%(R) for text class). Moreover, we show that these descriptors are more appropriate for the segmentation and characterization of graphic regions (high recall) than textual regions, but they label text pixels as belonging to the graphical class (low precision), *i.e.* the extracted descriptors produced an over-segmentation of the graphic regions. In this example, we obtain with the Gabor descriptors using the HAC algorithm 96%(PPB), 90%(CA), 91%(P), 84%(R) and 88%(F). On the other side, the second confusion matrix and pixel-clustering results using the k-means algorithm of a document containing graphics (green) and single text font (blue) is represented on the right. Similarly, we note that the extracted Gabor descriptors with the k-means algorithm tend to miss more text pixels than graphic pixels (46%(R) for text class). We obtain with the Gabor descriptors using the

k-means algorithm 92%(PPB), 73%(CA), 82%(P), 73%(R) and 77%(F). Thus, we conclude that the pixel-labeling results for the extracted Gabor features obtained with the HAC algorithm show a much greater discriminating power for separating text (single font) and graphic regions.

– The second example (*cf.* Figure 4.23) presents two confusion matrices and pixel-clustering results with the Gabor descriptors using the HAC and k-means algorithms of a document containing graphics and text with two different fonts. The first confusion matrix and pixel-clustering results using the HAC algorithm of a document containing graphics (green) and text with two different fonts (*"Font 1"*: text with S_1^f size font (red) and *"Font 2"*: text with $S_2^f \geq S_1^f$ size font (blue)) is represented on the left. In this example, the *"Font 1"* textual pixels are classified with 99.3%(P) and 99.9%(R), while for the *"Font 2"* textual pixels the values are 100%(P) and 100%(R). On the other side, the graphical pixels are classified with 100%(P) and 87%(R). Hence, the *"Font 2"* textual pixels class have a higher recall and a lower precision. Thus, we note that the extracted Gabor descriptors are more relevant for the segmentation and characterization of textual regions having higher size font (high recall) than textual regions having lower size font, but they label the pixel of textual regions having lower size font as belonging to the graphic class (low precision), *i.e.* the extracted Gabor descriptors with the HAC algorithm produce an over-segmentation of the textual regions having lower size font. So this confirms our hypothesis that the Gabor attributes provide better results for distinguishing different text fonts if the HDI under consideration contains only text. Moreover, we confirm the limitations of the Gabor approach to separate spatially close distinct kinds of information (*i.e.* the vertical/horizontal spacing is too small). In this example, we obtain with the Gabor descriptors using the HAC algorithm 98%(PPB), 99%(CA), 99%(P), 95%(R) and 97%(F). On the other side, the second confusion matrix and pixel-clustering results using the k-means algorithm of a document containing graphics and text with two different fonts. The first confusion matrix and pixel-clustering results using the HAC algorithm of a document containing graphics (blue) and text with two different fonts (*"Font 1"*: text with S_1^f size font (green) and *"Font 2"*: text with $S_2^f \geq S_1^f$ size font (red)) is represented on the right. Similarly, we note that the extracted Gabor descriptors with the k-means algorithm produce an over-segmentation of the textual regions having lower size font. We obtain with the Gabor descriptors using the k-means algorithm 98%(PPB), 98%(CA), 98%(P), 93%(R) and 96%(F). Thus, we conclude that the pixel-labeling results for the extracted Gabor features obtained with the HAC algorithm show a quite relatively similar discriminating power for separating text (two fonts) and graphic regions.

In order to get a better idea of how the extracted Gabor features are structured for each example, Euclidean distances (ED) between each pair of mean cluster feature values are illustrated at the bottom of the confusion matrix. The Euclidean distance ($ED(x, y)$) of two multivariate vectors $x = (x_1, x_2, \ldots, x_{N_f})^T$ and $y = (y_1, y_2, \ldots, y_{N_f})^T$ is defined as:

$$ED(x, y) = \sqrt{\sum_{i=0}^{N_f} (y_i - x_i)^2} = \sqrt{SED(x, y)} \tag{4.5}$$

A high distance ($d_{AB} = 1.43$) when using the HAC algorithm is noted for the first example for a document containing graphics and single text font (*cf.* Figure 4.22(a)), while a low one ($d_{CD} = 1.04$) when using the k-means algorithm (*cf.* Figure 4.22(b)). This strengthens our previous observations and confirms our assumption that the pixel-clustering results for the extracted Gabor features obtained with the HAC algorithm show a much greater discriminating power for separating text (single font) and graphic regions. Nevertheless, for the second example for a document containing graphics and two text fonts the distances between each text font and graphic classes when using the HAC algorithm (*cf.* Figure 4.23(a), $d_{AB} = 2.48$ and $d_{AC} = 2.10$) and those when using the k-means algorithm (*cf.* Figure 4.23(b), $d_{CD} = 2.08$ and $d_{CE} = 2.49$) are relatively similar. Moreover, the distances between two text font classes when using the HAC algorithm (*cf.* Figure 4.23(a), $d_{BC} = 1.51$) and those when using the k-means algorithm (*cf.* Figure 4.23(b), $d_{DE} = 1.53$) are quite similar. Therefore, we observe through the analysis of some HDI examples that if the HDI

under consideration contains graphics and two text fonts, the use of either the HAC algorithm or the k-means one in the pixel-clustering task of the proposed Gabor-based pixel-labeling scheme leads quite similar pixel-labeling results.

4.5.2.3. Quantitative results

By comparing the two clustering methods, k-means and HAC, on the auto-correlation and Gabor-based pixel-labeling approaches, higher performances are obtained by using the HAC algorithm (*cf.* Tables 4.6 and 4.7 and Figure 4.20 and 4.21).

The comparison results given by the two clustering algorithms, HAC *vs.* k-means, and the two texture-based feature sets, the auto-correlation *vs.* Gabor features, in the proposed pixel-labeling scheme on the DIGIDOC-Texture dataset, are presented in Tables 4.6 and 4.7, respectively. Table 4.8 shows the differences in the computed clustering and classification accuracy measures when using the HAC and k-means clustering algorithms and the two texture-based feature sets, the auto-correlation and Gabor in the proposed pixel-labeling scheme on the *"DIGIDOC-Texture dataset"*. Several clustering and classification accuracy measures are computed, J, PPB, P, R, F and CA, to evaluate quantitatively the obtained results given by the HAC *vs.* k-means clustering algorithms in the pixel-clustering task of the proposed pixel-labeling scheme, with the auto-correlation and Gabor features on the *"DIGIDOC-Texture dataset"*.

1. ***Auto-correlation features***:
   We observe that the two best average performances with using the auto-correlation features for most of the computed evaluation metrics are obtained for the *"One font and graphics"* and *"Two fonts and graphics**"* categories of the *"DIGIDOC-Texture dataset"* for the HAC (91%(PPB), 83%(P), 81%(R), 82%(F) and 85%(CA) for the *"One font and graphics"* HDI category, and 90%(PPB), 84%(P), 83%(R), 83%(F) and 84%(CA) for the *"Two fonts and graphics**"*), and k-means (84%(PPB), 80%(P), 78%(R), 78%(F) and 79%(CA) for the *"One font and graphics"* HDI category, and 90%(PPB), 83%(P), 82%(R), 82%(F) and 83%(CA) for the *"Two fonts and graphics**"*) clustering algorithms.

   We note the two worst average performances with using the auto-correlation features for most of the computed evaluation metrics are obtained for the *"Two fonts and graphics*"* and *"only three fonts"* categories of the *"DIGIDOC-Texture dataset"* for the HAC (83%(PPB), 59%(P), 60%(R), 59%(F) and 70%(CA) for the *"Two fonts and graphics*"* HDI category, and 77%(PPB), 63%(P), 61%(R), 61%(F) and 62%(CA) for the *"only three fonts"*), and k-means (79%(PPB), 59%(P), 59%(R), 59%(F) and 66%(CA) for the *"Two fonts and graphics*"* HDI category, and 69%(PPB), 56%(P), 54%(R), 55%(F) and 61%(CA) for the *"only three fonts"*) clustering algorithms.

   When using *"Two fonts and graphics**"* in computing *"Overall**"*, we also observe that the performance of the extracted auto-correlation descriptors is much better then when using *"Two fonts and graphics*"* in computing *"Overall*"*, *i.e.* overall performance gains of 2%(PPB), 6%(P), 5%(R), 6%(F) and 3%(CA) when using the HAC algorithm, and 3%(PPB), 6%(P), 5%(R), 5%(F) and 5%(CA) when using the k-means algorithm are noted. With the HAC algorithm, the overall results when using the auto-correlation features are quite encouraging since we obtain 84%(PPB), 70%(P), 69%(R), 69%(F) and 74%(CA) in computing *"Overall*"*, and 86%(PPB), 76%(P), 74%(R), 75%(F) and 77%(CA) in computing *"Overall**"*. On the other side, with the k-means algorithm, the overall results when using the auto-correlation features are also quite satisfying since we obtain 78%(PPB), 68%(P), 65%(R), 66%(F) and 70%(CA) in computing *"Overall*"*, and 81%(PPB), 74%(P), 70%(R), 71%(F) and 75%(CA) in computing *"Overall**"*.

   Thus, we state that the pixel-labeling results for the extracted auto-correlation features obtained with the HAC algorithm show a much greater discriminating power for separating

text (single font) and graphic regions than for distinguishing graphics and two or more text fonts or documents containing only three fonts. This can be explained by the fact that the auto-correlation attributes generally provide the main orientation of a texture (horizontal orientation for textual regions, while many orientations are present to different extents in graphic blocks). Thus, the auto-correlation descriptors behave better on text/graphic discrimination than on text fonts separation.

2. ***Gabor features***:
Similarly, we observe that the two best average performances with using the Gabor features for most of the computed evaluation metrics are obtained for the *"One font and graphics"* and *"Two fonts and graphics**"* categories of the *"DIGIDOC-Texture dataset"* for the HAC (96%(PPB), 90%(P), 86%(R), 88%(F) and 87%(CA) for the *"One font and graphics"* HDI category, and 98%(PPB), 91%(P), 88%(R), 89%(F) and 89%(CA) for the *"Two fonts and graphics**"*), and k-means (94%(PPB), 91%(P), 83%(R), 86%(F) and 86%(CA) for the *"One font and graphics"* HDI category, and 95%(PPB), 88%(P), 84%(R), 86%(F) and 86%(CA) for the *"Two fonts and graphics**"*) clustering algorithms.

We note the worst average performances with using the Gabor features for most of the computed evaluation metrics are obtained for the *"only three fonts"* category of the *"DIGIDOC-Texture dataset"* for the HAC (88%(PPB), 67%(P), 62%(R), 64%(F) and 68%(CA)), and k-means (85%(PPB), 65%(P), 60%(R), 62%(F) and 66%(CA)) clustering algorithms.

Moreover, we observe that the performances with using the Gabor features for the *"Two fonts and graphics*"* and *"Only two fonts"* categories of the *"DIGIDOC-Texture dataset"* are quite good for both the HAC (93%(PPB), 70%(P), 66%(R), 67%(F) and 75%(CA) for the *"Two fonts and graphics*"* HDI category, and 94%(PPB), 89%(P), 81%(R), 84%(F) and 82%(CA) for the *"Only two fonts"*) and k-means (89%(PPB), 68%(P), 64%(R), 65%(F) and 73%(CA) for the *"Two fonts and graphics*"* HDI category, and 89%(PPB), 84%(P), 73%(R), 77%(F) and 75%(CA) for the *"Only two fonts"*) algorithms compared to the results obtained with the Gabor features.

Therefore, we note that the Gabor features perform slightly better for both the HAC and k-means algorithms when the HDI under consideration contains a single text font and graphics or in the case of the analyzed HDI containing two fonts and graphics where all fonts in the text have the same label in the ground-truth, than when documents contains only three fonts or two fonts and graphics where every font in the text has a different label in the ground-truth. However, the overall results given by the Gabor features with both the HAC and k-means algorithms are better than of using the auto-correlation features on the pixel-clustering task of the proposed texture-based pixel-labeling scheme, particularly for the *"Two fonts and graphics*"*, *"Only two fonts"*, and *"Only three fonts"* categories of the *"DIGIDOC-Texture dataset"*. This strengthens our previous results and confirms our assumption that the Gabor descriptors are more suitable for font segmentation, since they are known to be sensitive to the stroke width.

When using *"Two fonts and graphics**"* in computing *"Overall**"*, we also observe that the performance of the extracted Gabor descriptors is much better then when using *"Two fonts and graphics*"* in computing *"Overall*"*, *i.e.* overall performance gains of 1%(PPB), 5%(P), 5%(R), 5%(F) and 4%(CA) when using the HAC algorithm, and 2%(PPB), 5%(P), 5%(R), 5%(F) and 4%(CA) when using the k-means algorithm are noted. With the HAC algorithm, the overall results when using the Gabor features are encouraging since we obtain 93%(PPB), 79%(P), 74%(R), 76%(F) and 78%(CA) in computing *"Overall*"*, and 94%(PPB), 84%(P), 79%(R), 81%(F) and 82%(CA) in computing *"Overall**"*. On the other side, with the k-means algorithm, the overall results when using the Gabor features are also quite satisfying since we obtain 89%(PPB), 77%(P), 70%(R), 73%(F) and 75%(CA) in

computing *"Overall*"*, and 91%(PPB), 82%(P), 75%(R), 78%(F) and 79%(CA) in computing *"Overall**"*.

3. ***Gabor vs. auto-correlation features and HAC vs. k-means clustering algorithms***: In Table 4.8, we note the performance differences in the computed accuracy metrics when using the HAC and k-means clustering algorithms with the auto-correlation and Gabor features. Overall, the difference values between the results using the HAC and k-means clustering algorithms are positive. The overall performance gains when using the HAC and k-means clustering algorithms for the Gabor features are: 8%(J), 5.2%(PPB), 1.9%(P), 3.7%(R), 2.8%(F) and 3.3%(CA) for *"Overall*"*, and 7.5%(J), 4.4%(PPB), 1.9%(P), 3.8%(R), 2.8%(F) and 2.3%(CA) for *"Overall**"*. On the other side, the overall performance gains when using the HAC and k-means clustering algorithms for the auto-correlation features are: 6.5%(J), 3.7%(PPB), 2.0%(P), 3.8%(R), 3.1%(F) and 2.6%(CA) for *"Overall*"*, and 6.7%(J), while 3.5%(PPB), 2.3%(P), 4.2%(R), 3.4%(F) and 2.8%(CA) are noted for *"Overall**"*. As a consequence, with the two kinds of texture features, the auto-correlation and Gabor, using the two clustering algorithms indicates that better results are obtained with the HAC technique than with the k-means algorithm.

4.6. Discussion

We have evaluated both qualitatively and quantitatively the effectiveness of the extracted texture-based feature sets (Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor, Haar, Db3 and Db4) in the discrimination of the foreground layers of a HDI, particularly of text and graphics. By comparing the performance results of the experimental corpus containing two datasets, the *"DIGIDOC-Texture dataset"* and *"HBR2013 dataset"*, we conclude that the scalability of the nine evaluated texture-based feature sets has proved for both datasets, even if the *"HBR2013 dataset"* presents few limitations (e.g. binary HDIs, low resolution digitization, presence of copyright notices in many pages).

Nevertheless, the fundamental question is if these texture-based feature sets have been compared properly or not. We should point out that the main technological bottleneck is the definition of an accurate and objective ground-truth. Antonacopoulos *et al.* [136] stated that a direct comparison between several algorithms is tough and critical task for a variety of DIA applications due to the need for a realistic data and the high requirement for an adequate ground-truth as well as the use of a set of objective evaluation criteria. However, it is still hard to determine fairly the different HDI content types. An important issue can also be outlined which consists of the difficulty to take into account the noisy foreground cluster when defining the ground-truth in the case of degraded HDIs. An and Baird [351] stipulated that the pixel-wise classifiers rely on the accuracy of ground-truth annotations. Since the defined ground-truth is not a pixel-based one (*i.e.* it is defined by spatial boundaries of regions with labels). This highlights the need for a pixel-based ground-truth. This issue has been also reported by Kumar *et al.* [217] who outlined that the use of a zone-level ground-truth might have an influence on the accuracy of pixel-level approach and particularly the R measure.

In this work, the noise pixels have not been considered when defining our ground-truth. Nevertheless, to our knowledge there really is no defined pixel-based ground-truth of HDIs which takes account the noise pixels. It is not a straightforward task to define appropriate and objective ground-truth due to the characteristics of HDIs (e.g. page skew, superimposition of information layers, such as stamps, handwritten notes, noise, back-to-front interference). The first aspect of future work will be to use a new computer-aided ground-truthing environment editor for creating and manipulating automatically meta-data corresponding to regions of interest on HDIs under consideration (*i.e.* to generate a pixel-based ground-truth including the noise pixels).

It is worth noting that there is awareness that many factors (e.g. binary HDIs, low resolution digitization, presence of copyright notices in many pages, defined ground-truth, number of classes

defined in the ground-truth, used pixel-labeling scheme for comparing texture, type of used pre-processing stage, kind of used feature extraction technique) can influence the comparative study and experimental evaluation of a number of commonly and widely used texture features conducted in this chapter. Our goal in this chapter is to analyze properly texture features by raising issues related only to how these texture-based sets are compared with each other. We have planned to avoid all unnecessary biases caused by introducing a feature selection task, such as the methods based on the dimension reduction technique or a post-processing step by integrating a post-processing phase by taking into consideration the topological or spatial relationships (e.g. hierarchy, inclusion, neighborhood position). In addition, based on a review of the literature and after performing several experiments to choose the best configuration of the different techniques, we have made a first reasonable attempt as much as possible to carry out a properly and appropriate comparative study on HDIs by using a standard pixel-labeling scheme for evaluating and benchmarking texture features. We are interested in determining which texture methods are firstly well suited for segmenting graphical regions from textual ones, discriminating text in a variety of situations of different fonts and scales and secondly in finding a constructive compromise between the performance and computational cost.

4.7. Conclusion

This chapter has presented an experimental evaluation and benchmarking of a number of commonly and widely used texture features. This comparative study has been conducted on a large corpus of HDIs for the purpose of determining the performance of each texture-based feature set according to the DI content, *i.e.* segmenting graphical regions from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand. The experimental corpus (1100 pages of historical documents) is composed of two datasets, the *"DIGIDOC-Texture dataset"* and *"HBR2013 dataset"*. We have proved the scalability of nine evaluated texture-based feature sets (Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor, Haar, Db3 and Db4) for both datasets. Thus, a standard pixel-labeling scheme for evaluating and benchmarking texture features has been proposed in this chapter to compare nine texture-based feature sets.

This work has shown the effectiveness of the texture analysis approaches for historical DIA. Based on our experiments, we conclude that the auto-correlation, Gabor and Db4 features are the best choices for discriminating textual regions from graphical ones without taking into account the spatial relationships between pixels. However, when the numerical complexity and pixel-labeling performance are taken into account, the Gabor approach would be the better choice. Furthermore, the Gabor approach is a good choice for segmenting HDIs containing only textual regions with different fonts. 76%, 80% and 76% classification accuracy values are noted when the auto-correlation, Gabor and Db4 are used in the proposed pixel-labeling scheme for evaluating and benchmarking texture features, respectively. The results reported in this work provide a useful benchmark in terms of performance evaluation, texture vector dimensionality, memory requirements, processing time and complexity for current and future research efforts in historical DIA.

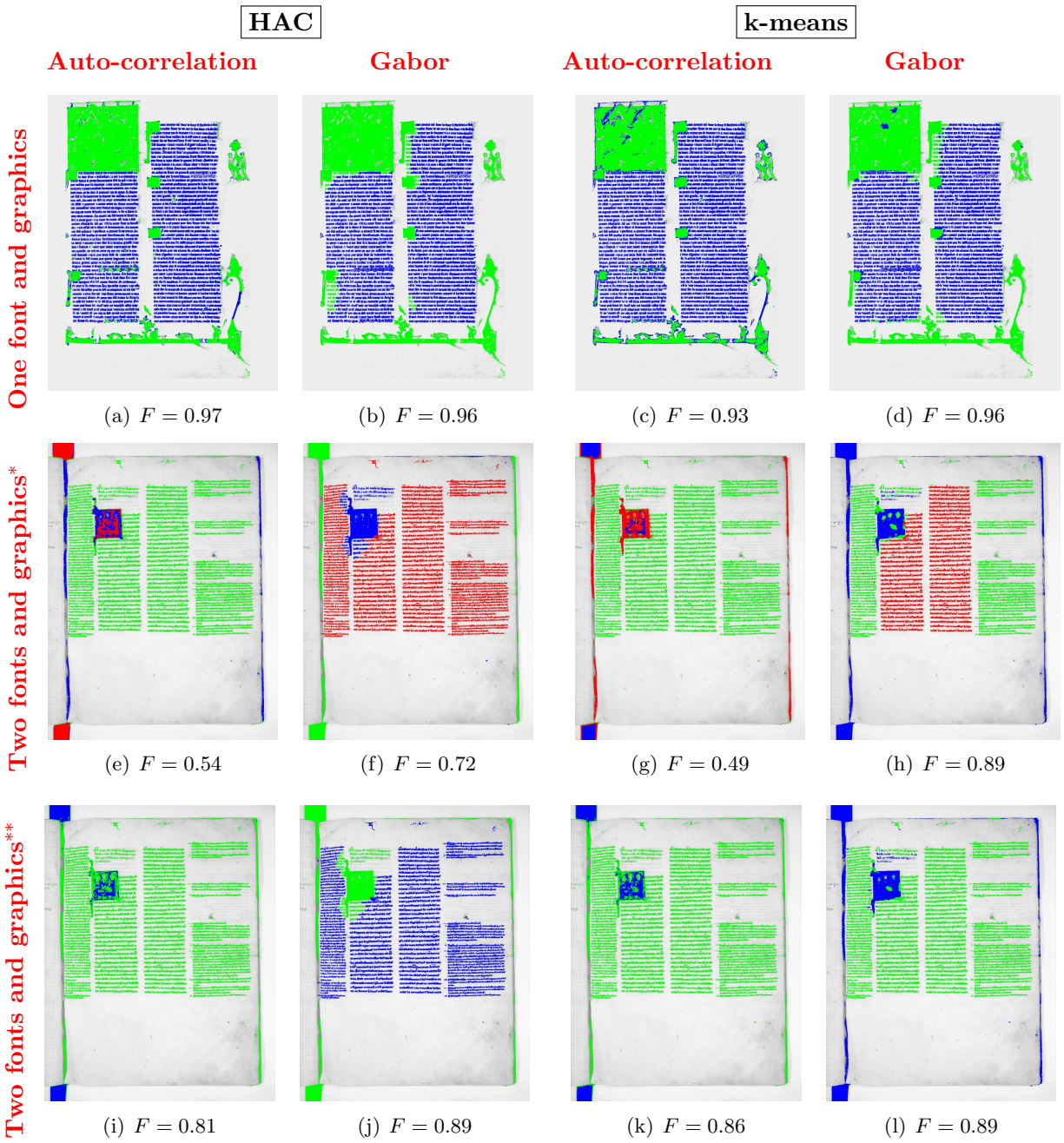
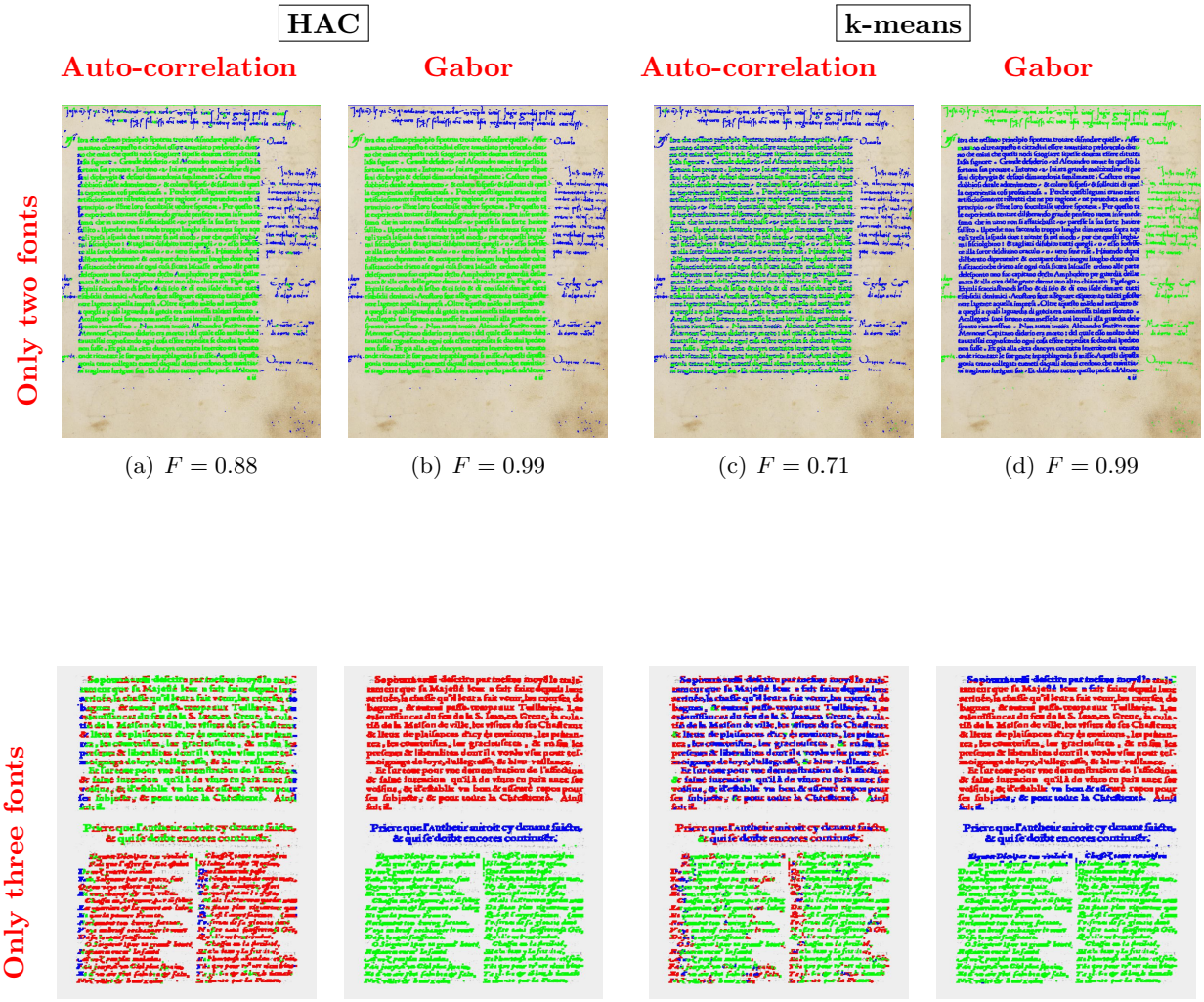


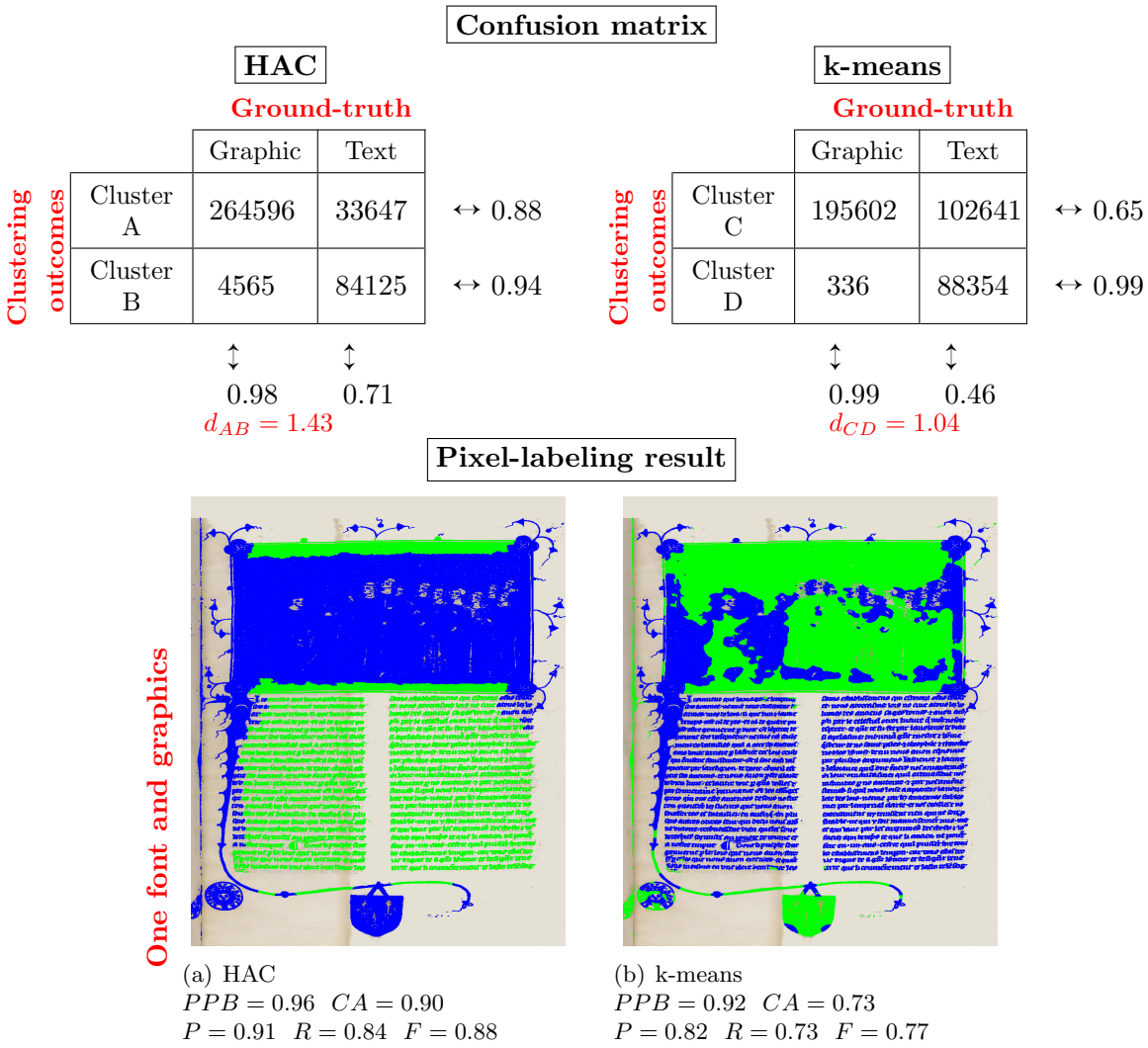
Figure 4.20.: Examples of resulting images of the clustering of the extracted texture features (**auto-correlation** and **Gabor**) from the ***"DIGIDOC-Texture dataset"*** (*"One font and graphics"*, *"Two fonts and graphics*"* and *"Two fonts and graphics**"*) using the HAC and k-means algorithms. The HAC and k-means algorithms are used with the normalized textural features by setting the maximum k clusters to that defined in the defined ground-truth. Figures (a), (b), (c) and (d) illustrate the different resulting HDIs of the clustering of the extracted auto-correlation and Gabor features using the HAC and k-means algorithms applied on an example of the *"One font and graphics"* category. Figures (e), (f), (g) and (h) illustrate the different resulting HDIs of the clustering of the extracted auto-correlation and Gabor features using the HAC and k-means algorithms applied on an example of the *"Two fonts and graphics*"* category. Figures (i), (j), (k) and (l) illustrate the different resulting HDIs of the clustering of the extracted auto-correlation and Gabor features using the HAC and k-means algorithms applied on an example of the *"Two fonts and graphics**"* category. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



(a) $F = 0.88$ (b) $F = 0.99$ (c) $F = 0.71$ (d) $F = 0.99$

(e) $F = 0.54$ (f) $F = 1.00$ (g) $F = 0.64$ (h) $F = 0.86$

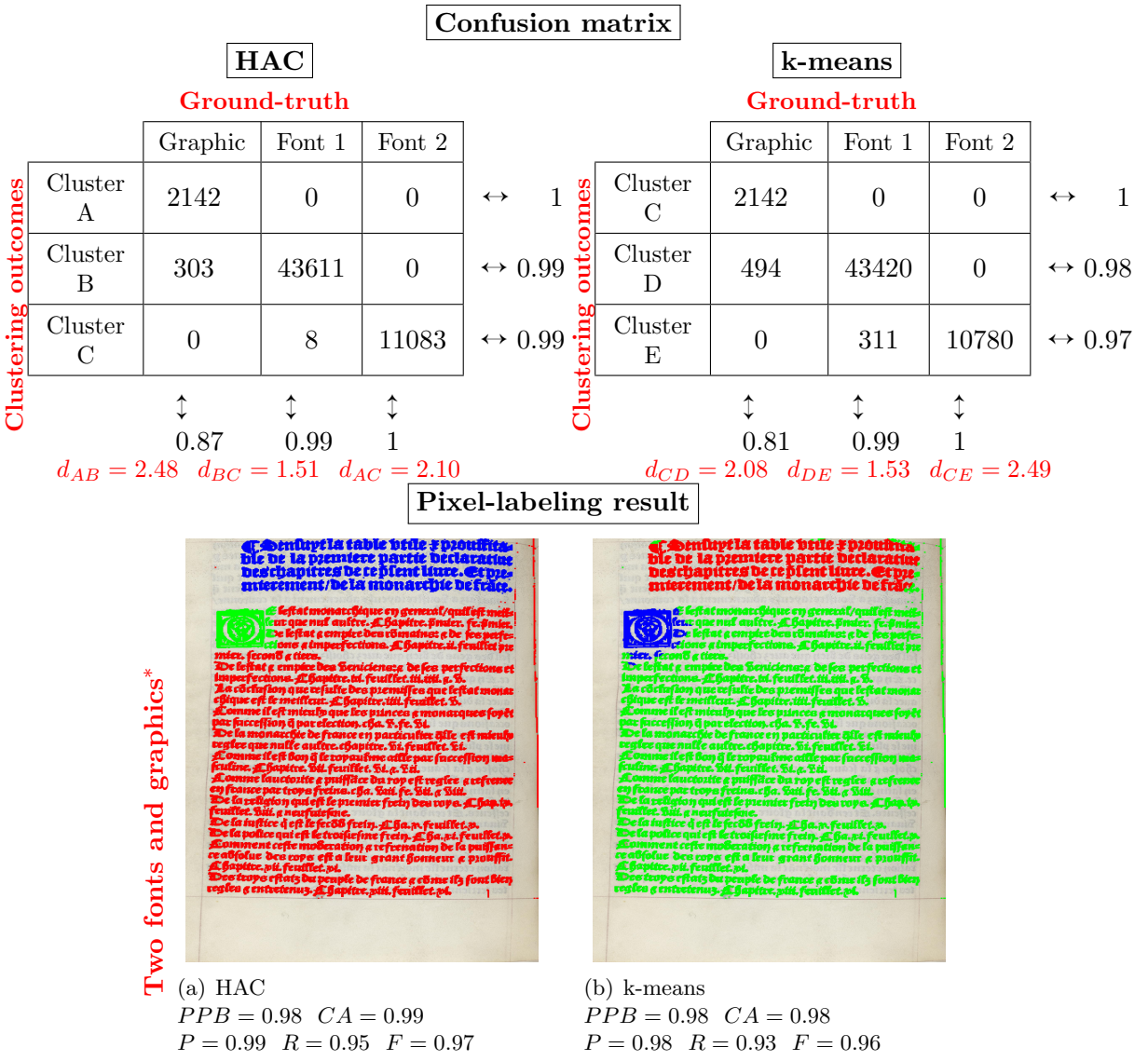
Figure 4.21.: Examples of resulting images of the clustering of the extracted texture features (**auto-correlation** and **Gabor**) from the ***"DIGIDOC-Texture dataset"*** (*"Only two fonts"* and *"Only three fonts"*) using the HAC and k-means algorithms. The HAC and k-means algorithms are used with the normalized textural features by setting the maximum k clusters to that defined in the defined ground-truth. Figures (a), (b), (c) and (d) illustrate the different resulting HDIs of the clustering of the extracted auto-correlation and Gabor features using the HAC and k-means algorithms applied on an example of the *"Only two fonts"* category. Figures (e), (f), (g) and (h) illustrate the different resulting HDIs of the clustering of the extracted auto-correlation and Gabor features using the HAC and k-means algorithms applied on an example of the *"Only three fonts"* category. Since the process is unsupervised, the colors attributed to text or graphics may differ from one DI to another.



(a) HAC
$PPB = 0.96$ $CA = 0.90$
$P = 0.91$ $R = 0.84$ $F = 0.88$

(b) k-means
$PPB = 0.92$ $CA = 0.73$
$P = 0.82$ $R = 0.73$ $F = 0.77$

Figure 4.22.: Examples of confusion matrix computation and pixel-labeling results of a document from the *"DIGIDOC-Texture dataset"*, containing graphics and single text font ***"One font and graphics"***, obtained using the HAC and k-means algorithms and by setting the maximum number of clusters to 2. Figure (a) represents the pixel-labeling result of a document containing graphics (blue) and single text font (green) using the HAC algorithm. Figure (b) the pixel-labeling result of a document containing graphics (green) and single text font (blue) using the k-means algorithm.



(a) HAC
$PPB = 0.98$ $CA = 0.99$
$P = 0.99$ $R = 0.95$ $F = 0.97$

(b) k-means
$PPB = 0.98$ $CA = 0.98$
$P = 0.98$ $R = 0.93$ $F = 0.96$

Figure 4.23.: Examples of confusion matrix computation and pixel-labeling results of a document from the *"DIGIDOC-Texture dataset"*, containing graphics and two different text fonts ***"Two fonts and graphics*"***, obtained using the HAC and k-means algorithms and by setting the maximum number of clusters to 3. Figure (a) represents the pixel-labeling result of a document containing graphics (green) and two different text fonts (blue and red) using the HAC algorithm. Figure (b) the pixel-labeling result of a document containing graphics (blue) and two different text fonts (green and red) using the k-means algorithm.

Table 4.6.: Evaluation of the extracted **auto-correlation features** by clustering and classification accuracy measures on the ***"DIGIDOC-Texture dataset"*** using the **HAC** and **k-means** algorithms: Jaccard coefficient (J), purity per block metric (PPB), precision (P), recall (R), F-measure (F) and classification accuracy (CA). $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results. The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
HAC	One font and graphics	0.79	0.17	0.91	0.08	0.83	0.16	0.81	0.17	0.82	0.16	0.85	0.21
	Two fonts and graphics*	0.61	0.17	0.83	0.09	0.59	0.11	0.60	0.12	0.59	0.11	0.70	0.19
	Two fonts and graphics**	0.70	0.17	0.90	0.07	0.84	0.15	0.83	0.16	0.83	0.14	0.84	0.19
	Only two fonts	0.71	0.18	0.84	0.11	0.73	0.16	0.72	0.17	0.72	0.15	0.78	0.23
	Only three fonts	0.61	0.19	0.77	0.10	0.63	0.13	0.61	0.13	0.61	0.12	0.62	0.25
	Overall*	**0.68**	**0.18**	**0.84**	**0.10**	**0.70**	**0.14**	**0.69**	**0.15**	**0.69**	**0.14**	**0.74**	**0.22**
	Overall**	**0.70**	**0.18**	**0.86**	**0.09**	**0.76**	**0.15**	**0.74**	**0.16**	**0.75**	**0.14**	**0.77**	**0.22**
	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
k-means	One font and graphics	0.69	0.18	0.84	0.13	0.80	0.15	0.78	0.18	0.78	0.16	0.79	0.19
	Two fonts and graphics*	0.54	0.15	0.79	0.11	0.59	0.10	0.59	0.11	0.59	0.10	0.66	0.16
	Two fonts and graphics**	0.65	0.16	0.90	0.08	0.83	0.14	0.82	0.16	0.82	0.14	0.83	0.16
	Only two fonts	0.66	0.18	0.81	0.13	0.75	0.16	0.69	0.16	0.71	0.15	0.75	0.19
	Only three fonts	0.51	0.17	0.69	0.12	0.56	0.11	0.54	0.11	0.55	0.10	0.61	0.18
	Overall*	**0.60**	**0.17**	**0.78**	**0.12**	**0.68**	**0.13**	**0.65**	**0.14**	**0.66**	**0.13**	**0.70**	**0.18**
	Overall**	**0.63**	**0.17**	**0.81**	**0.12**	**0.74**	**0.14**	**0.70**	**0.15**	**0.71**	**0.14**	**0.75**	**0.18**

Table 4.7.: Evaluation of the extracted **Gabor features** by clustering and classification accuracy measures on the ***"DIGIDOC-Texture dataset"*** using the **HAC** and **k-means** algorithms: Jaccard coefficient (J), purity per block metric (PPB), precision (P), recall (R), F-measure (F) and classification accuracy (CA). $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results. The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
HAC	One font and graphics	0.88	0.18	0.96	0.06	0.90	0.16	0.86	0.19	0.88	0.17	0.87	0.25
	Two fonts and graphics*	0.70	0.16	0.93	0.06	0.70	0.16	0.66	0.13	0.67	0.14	0.75	0.18
	Two fonts and graphics**	0.81	0.16	0.98	0.04	0.91	0.13	0.88	0.16	0.89	0.14	0.89	0.21
	Only two fonts	0.82	0.22	0.94	0.09	0.89	0.15	0.81	0.22	0.84	0.18	0.82	0.25
	Only three fonts	0.60	0.19	0.88	0.09	0.67	0.17	0.62	0.18	0.64	0.17	0.68	0.19
	Overall*	**0.75**	**0.19**	**0.93**	**0.08**	**0.79**	**0.16**	**0.74**	**0.18**	**0.76**	**0.17**	**0.78**	**0.22**
	Overall**	**0.78**	**0.19**	**0.94**	**0.07**	**0.84**	**0.15**	**0.79**	**0.19**	**0.81**	**0.17**	**0.82**	**0.23**
	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
k-means	One font and graphics	0.82	0.20	0.94	0.08	0.91	0.13	0.83	0.19	0.86	0.16	0.86	0.20
	Two fonts and graphics*	0.65	0.17	0.89	0.08	0.68	0.15	0.64	0.13	0.65	0.13	0.73	0.17
	Two fonts and graphics**	0.75	0.18	0.95	0.06	0.88	0.14	0.84	0.17	0.86	0.15	0.86	0.18
	Only two fonts	0.70	0.23	0.89	0.11	0.84	0.15	0.73	0.21	0.77	0.18	0.75	0.23
	Only three fonts	0.56	0.19	0.85	0.10	0.65	0.17	0.60	0.17	0.62	0.16	0.66	0.17
	Overall*	**0.68**	**0.20**	**0.89**	**0.09**	**0.77**	**0.15**	**0.70**	**0.17**	**0.73**	**0.16**	**0.75**	**0.19**
	Overall**	**0.71**	**0.20**	**0.91**	**0.09**	**0.82**	**0.15**	**0.75**	**0.18**	**0.78**	**0.16**	**0.79**	**0.19**

Table 4.8.: Differences in the computed clustering and classification accuracy measures when using the **HAC** and **k-means** algorithms in the **auto-correlation** and **Gabor**-based pixel-labeling scheme on the ***"DIGIDOC-Texture dataset"***: Jaccard coefficient (J), purity per block metric (PPB), precision (P), recall (R), F-measure (F) and classification accuracy (CA). The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	***J***	***PPB***	***P***	***R***	***F***	***CA***
Auto-correlation	One font and graphics	0.098	0.064	0.027	0.037	0.032	0.057
	Two fonts and graphics*	0.064	0.035	0.005	0.004	0.006	0.043
	Two fonts and graphics**	0.045	0.003	0.006	0.009	0.008	0.003
	Only two fonts	0.057	0.026	-0.022	0.034	0.006	0.027
	Only three fonts	0.101	0.085	0.065	0.073	0.067	0.006
	Overall*	**0.080**	**0.052**	**0.019**	**0.037**	**0.028**	**0.033**
	Overall**	**0.075**	**0.044**	**0.019**	**0.038**	**0.028**	**0.023**
	Document content	***J***	***PPB***	***P***	***R***	***F***	***CA***
Gabor	One font and graphics	0.060	0.028	-0.004	0.028	0.016	0.002
	Two fonts and graphics*	0.049	0.038	0.020	0.022	0.022	0.018
	Two fonts and graphics**	0.058	0.029	0.031	0.038	0.035	0.025
	Only two fonts	0.111	0.047	0.044	0.085	0.067	0.071
	Only three fonts	0.040	0.035	0.021	0.018	0.020	0.014
	Overall*	**0.065**	**0.037**	**0.020**	**0.038**	**0.031**	**0.026**
	Overall**	**0.067**	**0.035**	**0.023**	**0.042**	**0.034**	**0.028**

Chapter 5.

A texture-based pixel-labeling framework for digitized historical books

This chapter presents a framework to investigate the use of texture as a tool for determining automatically the number of content types (different text fonts and graphic regions) in a digitized historical book and segmenting its contents by extracting and analyzing texture features independently of the layout of the pages. The proposed framework is parameter-free and applicable to a large variety of ancient of books. It does not assume *a priori* information regarding document image layout and content.

Contents

5.1. Introduction

Over the last few years, there has been tremendous growth in the automatic processing of digitized HDIs. In fact, finding reliable systems for the interpretation of HDIs has been a topic of major interest for many libraries and the prime issue of research in the historical DIA community. One important challenge is to refine well-known approaches based on strong *a priori* knowledge (e.g. DI content, layout, typography, font size and type, scanning resolution, DI size). Nevertheless, a texture analysis approach has consistently been chosen to segment a page layout when information is lacking on DI layout and content.

For this purpose, we propose to characterize digitized pages of ancient books with a set of regions of homogeneous texture and their topological relationships that helps modeling the layout structure, separating text from non-text regions, partitioning or categorizing pre-localized text blocks into columns, headings, paragraphs, lines, words, notes (head-notes and foot-notes) and abstracts, *etc.* Our goal is to extract as automatically as possible textural features that segment an ancient book or a collection of HDIs into spatially disjoint homogeneous regions or similar content regions and characterize its content according to a topological representation of homogeneous regions, without formulating a hypothesis concerning the DI structure or layout (e.g. column layout), typographical parameters (e.g. font size and type) or graphical properties of the analyzed DI.

Recently, the issues of DIA have been considered as texture segmentation and classification [6]. Moreover, some similarities of HDI content type have been deduced from many book pages [11, 12]. In addition, based on the assumption that texture can characterize a HDI content type which is usually repeated on many pages of the same book, we propose a framework that works on entire book scale instead of processing each page individually. Thus, in this chapter by combining several points related to texture-based segmentation that have been reported separately in the literature particularly on synthetic, medical and natural images, we attempt to represent a book page using a set of homogeneous blocks defined by similar texture attributes and their topology. Indeed, a pixel-labeling framework for DHBs is proposed in this chapter. The proposed framework ensures the pixel-based characterization of the content of an entire book by extracting and analyzing the texture information from each page. It is automatic, parameter-free and can be adapted to all kinds of books. It is independent of DI layout, typeface, font size, orientation, image size, digitizing resolution and intensity, *etc.* It is also insensitive to noise. Moreover, it does not require any manual inspection or *a priori* knowledge regarding DI content and structure or layout.

The originality of our contribution lies in the automatic analysis of some characteristics of book pages (regarding their layout and/or content) to find the number of book content types (*i.e.* by identifying graphic and textual regions) by extracting and clustering texture features on an entire book instead of processing each page individually, with no assumption concerning the book page structure or layout (e.g. column layout), typographical characteristics or graphical properties (e.g. font size and type) of the digitized book pages. Indeed, even if the typographical or graphical features are not known in advance, they can be captured by exploiting the regularities of the associated textures through the whole book pages. So, in a first step, a clustering of texture features which are extracted from a sub-sampling in the entire book aims at identifying the texture information that is present in book pages. The clustering method that is applied has the ability to determine automatically the number of clusters or book content types. This knowledge is then used in a second step to segment each book page individually.

The remainder of this chapter is structured as follows: Section 5.2 reviews related works on texture-based approaches for DHBs. In Section 5.3, the proposed framework for the characterization of the content of an entire book by extracting and analyzing the texture information from each page is described. In Section 5.4, we outline the experimental protocol by describing the experimental corpus, the defined ground-truth and the used clustering and classification metrics for an evaluation of accuracy. Then, to evaluate the performance of the proposed framework, several clustering and classification metrics are computed and discussed. Qualitative results are also given to demonstrate its performance. Our discussion and conclusions are presented in Sections 5.5 and 5.6, respectively.

5.2. A short review of texture-based approaches for digitized historical books

A few texture-based segmentation approaches used with HDIs have been developed. To our knowledge, the only non-supervised texture-based approach used with DHBs was proposed by Journet *et al.* [1] It was based on an unsupervised clustering technique using extracted texture features which were computed from six pages of the same book. To assign the same label to pixels of six book pages which share similar textural characteristics, the clustering was performed on all extracted texture features of pixels of six book pages. They extracted two different kinds of texture descriptors for each pixel: three auto-correlation features which were derived from the rose of directions and two frequency attributes by using a multi-scale analysis for classifying HDI pixels into text, graphics and background. The first frequency descriptor computes the ink/paper transitions obtained by performing the average per-line sum of the difference between the pixel intensity value and its left neighbor. The frequency second attribute calculates the white spaces obtained by performing the RXYC algorithm and computing the mean of the average per-line and per-column sums of pixel intensities over an analyzed area. Then, by using the CLARA algorithm, an unsupervised clustering algorithm, the extracted texture descriptors were clustered and pixels were separated into different content clusters. Moreover, the number of book content types was assumed to be known in advance. They noted 83% and 92% mean good classification rates for the graphical and text pixels, respectively with 180 minutes in total per HDI as time required to process a page (feature extraction and pixel-clustering tasks) [3].

However the Journet *et al.*'s texture-based approach [1] yields good results on HDIs containing several textural classes (e.g. text, graphics, background), one main disadvantage is that there is a need of user intervention for setting the number of expected clusters, as a consequent of using a classical unsupervised clustering. Then, the most serious disadvantages of their approach is its high computational cost caused by the texture feature extraction step which was processed on all page pixels (*i.e.* foreground and background pixels). Finally, to assign the same label to pixels of six book pages which shares similar textural characteristics, the clustering approach was only performed on six pages of the same book instead of all book pages which can lead assigning different labels to each resulting cluster of two different sets of six pages of the same book (characterizing by similar textural properties).

Thus, in this chapter a texture-based pixel-labeling framework is proposed for the segmentation and characterization of DHB content which addresses the challenges of the existing state-of-the-art methods [1].

5.3. Proposed texture-based pixel-labeling framework

For the segmentation and characterization of DHB content, our goal is to determine a region or group of pixels which share similar properties or characteristics on the basis of which they are grouped. These characteristics may be based on the localization of the pixels and their surroundings, color, intensity or texture. In this chapter, we will focus only on texture-based features. The use of a texture-based approach in this work has been shown to be effective with skewed and degraded images [228]. We propose a framework which automatically extracts texture descriptors and involves a multi-resolution/multi-scale approach. This approach can characterize the content of DHB. In particular, it can discriminate between the different classes of the foreground layers of a digitized DI based on texture descriptors. The extraction of texture-based features helps to describe the DI layout and structure by analyzing the texture feature space computed from DHB content, *i.e.* by mapping the differences in the spatial structures of digitized DIs into differences in gray value for each page. Texture features are automatically extracted from the analyzed DI at several resolutions. The extracted features are then used in a parameter-free unsupervised clustering approach to determine the number of book content types that are defined by similar textural

descriptors. The proposed framework is pixel-based and does not require *a priori* knowledge of the DI layout, typographical parameters or graphical properties of the analyzed DI. Moreover, the number of homogeneous or similar content regions do not need to be known in advance as it is automatically determined. Thus, this framework is automatic, parameter-free and applicable to a large variety of DHBs. It is independent of DI layout, typeface, font size, orientation, digitizing resolution, *etc.* Moreover, it does not require any manual inspection.

The originality of this framework lies in the texture feature analysis that is used to find the number of book content types by utilizing a clustering approach on an entire book instead of processing each page individually. The proposed framework is supported by the fact that pages of the same book usually present strong similarities in the organization of the DI information (*i.e.* book page layout or structure) and in the graphical (e.g. embellishment, engraving, pictures) and typographical (e.g. font size and type) features throughout the DHB pages. Indeed, the texture information (e.g. typographical or graphical properties) which is often repeated and recurrently present in many book pages, can be deduced by exploiting the regularities of the associated textures through the whole book pages. Thus, a clustering step is performed on texture features which are extracted from a sub-sampling in the entire book aims at identifying these book characteristics which are then used to help to characterize each page image.

The proposed framework starts with a texture feature extraction step. Secondly, a number of foreground pixels from pages of the same book are selected randomly, and their textural descriptors are subsequently extracted in order to estimate the number of homogeneous or similar content regions in the book. The estimated number of book content types in the samples of foreground pixels is automatically determined (*cf.* block 2, Figure 5.1). Finally, the textural features for each page are used in a clustering approach by taking into account the estimation of the number of book content types (*cf.* block 1, Figure 5.1).

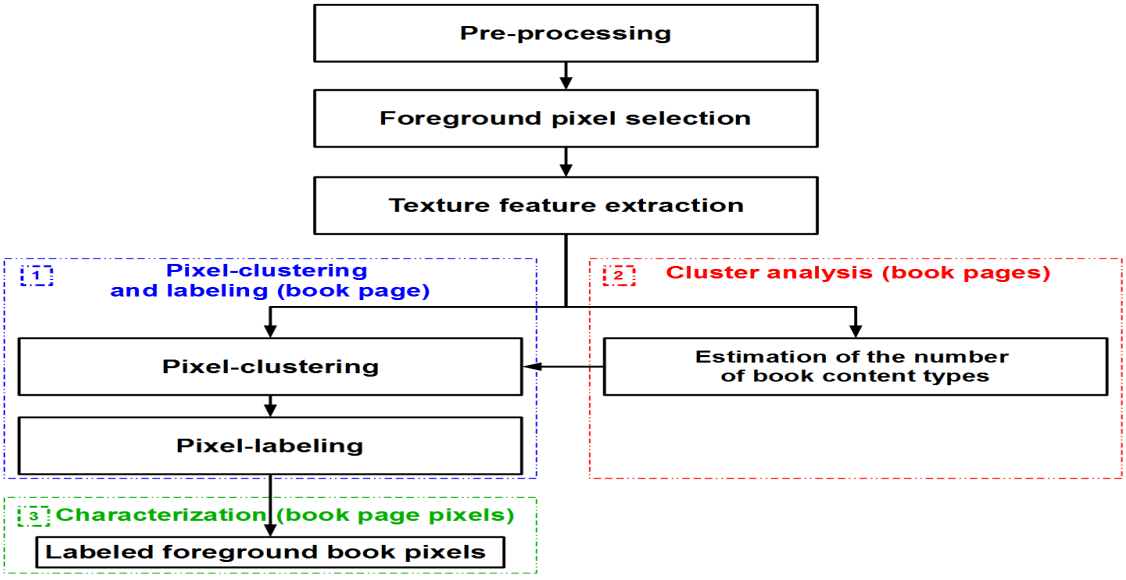


Figure 5.1.: Flowchart of the proposed texture-based pixel-labeling framework of DHB content.

Figure 5.1 illustrates the four main tasks of the proposed framework. Block 2 on Figure 5.1 is used to estimate the number of different book content types from the extracted textural features analyzed in the whole book. Block 1 on Figure 5.1 integrates an unsupervised task which automatically labels content pixels with the same cluster identifier as used with the book content in order to characterize the page foreground pixels of the digitized book (*cf.* block 3, Figure 5.1).

Figure 5.2 illustrates the detailed schematic block representation of the proposed texture-based pixel-labeling framework of DHB content.

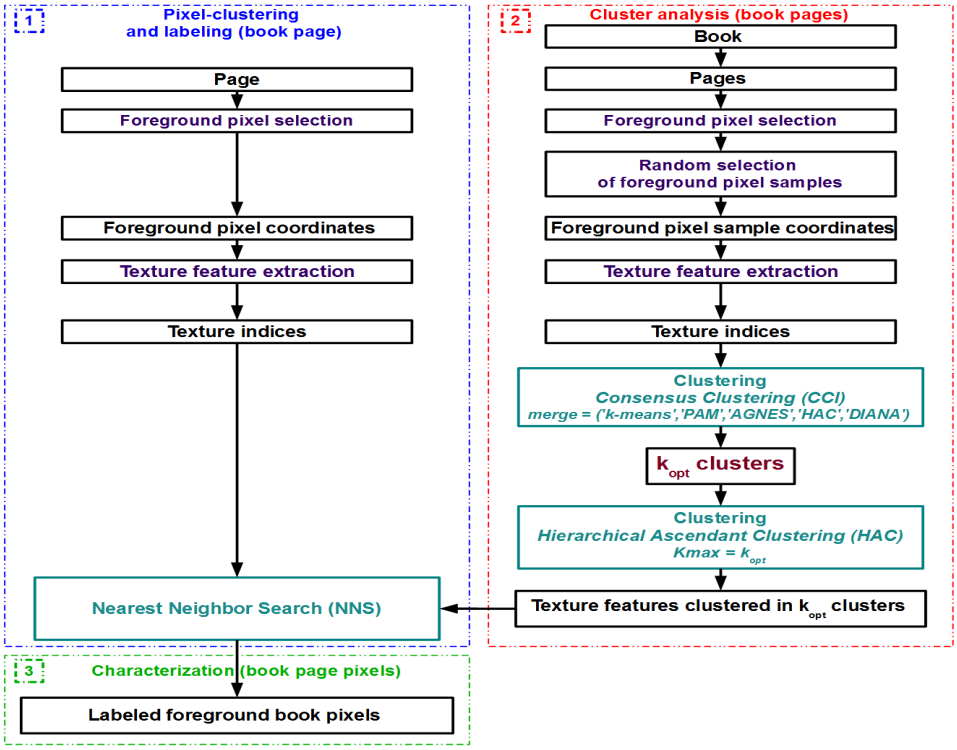


Figure 5.2.: Detailed schematic block representation of the proposed texture-based pixel-labeling framework of DHB content.

The proposed texture-based pixel-labeling framework of DHB content consists of the following four tasks:

1. ***Pre-processing and foreground pixel selection*** (*cf.* Section 4.4.1.1),
2. ***Texture feature extraction*** (*cf.* Section 4.4.1.2),
3. ***Estimation of the number of book content types*** (*cf.* Section 5.3.1),
4. ***Pixel-clustering and labeling*** (*cf.* Section 5.3.2).

The two first stages of the proposed texture-based pixel-labeling framework of DHB content, the pre-processing and foreground pixel selection, and the texture feature extraction have been previously described in Sections 4.4.1.1 and 4.4.1.2, respectively. These tasks aim to provide textural indices in the form of a set of features from the analyzed image to represent and characterize its content. Then, the next framework task consists in structuring the texture feature space within a hierarchical or partitioning clustering technique in order to group pixels sharing similar characteristics to identify and characterize similar regions or groups of pixels.

5.3.1. Estimation of the number of book content types

As already seen on the framework figure (*cf.* Figure 5.1), our objective is to find the number of book content types defined by similar texture features. So at this stage (*cf.* block 2, Figure 5.1) we need to use a clustering algorithm to partition the analyzed document into regions with similar properties or characteristics as deduced from the analysis of the extracted texture features presented in Section 4.4.1.2. With the help of the consensus clustering (CCl) technique (*cf.* Section 5.3.1.2), we automatically estimate the number of clusters from a number of samples of foreground pixels to determine the number of book content types defined by similar texture indices in a DHB.

5.3.1.1. Related works

For a certain class of hard clustering algorithms and particularly conventional clustering techniques [352, 353, 354, 355], the number of clusters in a dataset must be specified. Several types of methods can be used to estimate the correct number of clusters. The elbow method analyzes the percentage of variance explained as a function of the number of clusters [356]. The gap statistic evaluates the change in within-cluster dispersion [357]. Another set of techniques used to estimate the number of clusters is the information criteria approach [358], including Akaike information criterion (AIC) [359], Bayesian information criterion (BIC) [360] and integrated completed likelihood (ICL) [361]. Moreover, the *a priori* theory may be handled as a non-statistical technique for determining the number of clusters [362]. A theoretical approach using non-parametric information to choose the number of clusters based on distortion has been presented [363]. Some authors proposed a clustering methodology based on the cover coefficient concept that determines the number of clusters within a document database and relates indexing and clustering analytically [364]. Others defined the optimal number of clusters on the basis of the minimum description length (MDL) principle computed from the kernel matrix [365]. Furthermore, a visual diagnostic tool for choosing the number of clusters has been proposed [366]. Additionally, a link-based cluster ensemble framework was used to select the correct number of clusters after evaluating the clustering result of a variety of functional methods based on both internal and external criteria [367]. The cubic clustering criterion (CCC) is a measure of within-cluster homogeneity relative to between-cluster heterogeneity. The appropriate number of clusters is indicated by a peak in the CCC [368]. Ray and Turi [369] determined the number of clusters for color image segmentation in a clustering algorithm by finding the minimum of the intra-cluster and inter-cluster distance. An approach to determine the cluster boundaries in the hierarchical clustering based on within-class variance and between-class variance has been reported in [370]. Another technique has been proposed for the analysis of changes in silhouette values computed from clusters built by using the k-means algorithm and an optimization technique such as genetic algorithms [371]. Moreover, the v-fold cross-validation applied to a clustering algorithm, was performed for a range of numbers of clusters in the k-means or EM clustering [372]. Then, depending on the average distance of the observations (in the cross-validation or testing samples) from their cluster centers (for the k-means clustering), the number of clusters was estimated. Otherwise, by varying all combinations of the number of clusters, distance measures and clustering methods, the changes in various clustering evaluation indices can be examined [373]. Kryszczuk and Hurley [374] proposed a framework for the estimation of the numbers of clusters based on the decision-level fusion technique of multiple clustering validity indices. They proved that no single clustering validity indice consistently outperforms others, particularly for high-dimensional datasets. Bolshakova and Azuaje [375] proposed a weighed voting technique based on three clustering algorithms and two cluster validation indices to improve the prediction of the number of clusters.

Several clustering evaluation metrics for the estimation of the number of clusters and the criteria used to select the optimal number of clusters are summarized in Appendix A and particularly in Section A.3 (*cf.* Table A.4). For example, the maximum value of the Krzanowski-Lai index, Calinski-Harabasz index, silhouette width index and CCC are taken as indicating the correct number of clusters in the dataset, while the minimum value of C-index, Davies-Bouldin index, SDindex and SDbw validity index denote the correct number of clusters. Then, the criteria used to select the optimal number of clusters for the Hartigan index and Scott index is the maximum difference between hierarchy levels of the index [376].

5.3.1.2. Consensus clustering

Previous work identified a number of approaches for determining the correct number of clusters in a dataset [356]. Simpson *et al.* [377] have recently proposed an effective method, known as the CCl technique, to estimate the optimal number of clusters in biological data. Thus, we use the CCl in

this work to estimate the number of book content types (*i.e.* the number of similar texture-content types).

The CCl technique consists in performing a consensus matrix by iterating multiple runs of clustering algorithms with random and re-sampled clustering options [378]. Thus, the consensus matrix analyzes the consistency of the clustering result from five different clustering algorithms: AGNES, DIANA, PAM, k-means and HAC (*cf.* Appendix A and particularly Section A.1). So, by weighting the different clustering methods in order to mitigate extremes in the consensus values that can result from the sensitivity of some algorithms, a merge consensus matrix is performed which ensures the stability of the obtained clusters. Finally, the optimal number of clusters corresponds to the largest change in area under the cumulative density curve for the merge consensus matrix. It has been shown that the hierarchical clustering methods are highly sensitive to outliers while the partitioning ones are relatively insensitive. Simpson *et al.* [377] therefore used a merged CCl technique by applying a weighted averaging of the clustering result to estimate the number of clusters.

Thus, the number of clusters in a set of randomly selected foreground pixels is estimated from a few randomly selected pages of a book using the CCl method. This method is only used for a set of randomly selected foreground pixels of a few pages selected randomly from the same book. Due to memory constraints and long computational time of the CCl method, we first test it on a set of 1000 and 2000 randomly selected foreground pixels from few pages selected randomly from the same book.

Variations in clustering for both the hierarchical clustering and partitioning methods can be taken into consideration by associating non-uniform weights. With this approach, prior information is introduced into the clustering process by assigning higher weights to the most robust clustering methods. Thus, by weighting different clustering methods, extremes are mitigated in consensus values that can be created by the sensitivity of some algorithms, meaning that outliers can be dealt with differently within datasets, thus improving the quality of classification. So a weight of $\frac{1}{8}$ is assigned to each hierarchical clustering method (AGNES, DIANA and HAC), and a higher weight of $\frac{1}{4}$ is assigned to each partitioning clustering algorithm (PAM and k-means) [377]. By using this merge CCl technique, the consensus matrices are the results deduced from clustering experiments using different algorithms and/or conditions. The merging of clustering result between different methods provides an averaged clustering robustness, *i.e.* a merge consensus matrix M_{mc}. Hence, the optimal number of clusters k_{opt} in a dataset can be estimated by finding the value of k computed from the merge consensus matrix M_{mc} across a range $[2, 10]$ of possible values of k. The cumulative density function ($CDF(c)$) is computed on the unique elements of the merge consensus matrix M_{mc} sorted in descending order and defined over the range $c = [0, 1]$. Thus, the $CDF(c)$ is defined using equation 5.1.

$$CDF(c) = \frac{\sum\limits_{i<j} \mathbb{1}_{M(i,j)\leq c}}{\frac{N_s(N_s-1)}{2}} \tag{5.1}$$

where N_s is the number of selected observations or samples and $\mathbb{1}$ is an indicator or a characteristic function defined on a set $M_{mc}(i, j) \leq c$.

The area under the cumulative density curve (AUC) is then computed from the CDF (*cf.* equation 5.1) of the consensus matrix across a range $[2, 10]$ of possible values of k using equation 5.2.

$$AUC = \sum_{i=2}^{m} [y_i - y_{i-1}]\, CDF(y_i) \tag{5.2}$$

where y_i is the current element of the CDF and m is the number of elements of the CDF.

Finally, the optimal number of clusters k_{opt} corresponds to the largest change (Δk) in the AUC, where Δk denotes the difference change between two consecutive elements k in the AUC(*cf.* equation 5.2).

5.3.2. Pixel-clustering and labeling

Since the feature extraction phase and the task of the estimation of the optimal number of homogeneous or similar texture content types (k_{opt}) have been performed, we need to characterize the content of an entire book and to find the k_{opt} book content types defined by similar texture indices in a whole book. The goal of the third task of the proposed framework (*cf.* block 1, Figure 5.1) is to structure the texture feature space within a hierarchical or partitioning clustering technique in order to group pixels sharing similar characteristics to identify and characterize similar regions or groups of pixels.

5.3.2.1. Pixel-clustering

In this work, we opt for a standard and reliable hard clustering algorithm, given its optimal trade-off between low complexity, accuracy of the results, reduced number of parameter settings and the requirement for a clustering technique. This stage (*cf.* block 1, Figure 5.1) consists in grouping automatically the pixels into k_{opt} clusters representing homogeneous or similar texture-content regions. Since the main purpose of the CCl technique is to compare, visualize and evaluate the repeatability of the results of clustering experiments, and given the high demand in terms of memory and computational time of the CCl algorithm, we perform the HAC algorithm on the computed texture features without taking into account the spatial coordinates to search and extract similar texture-content pixels for each digitized book page (*i.e.* by grouping foreground pixels having similar page content type). The HAC algorithm process has been previously detailed in Chapter 4 and particularly Section 4.4.1.3.

The texture feature vectors are normalized to zero mean and unit standard deviation in order to avoid a domination of the higher numerical range of a few features. By setting the maximum number of book content types to the k_{opt} which is estimated with the CCl method, the adapted HAC algorithm with the Ward criterion can be applied to the normalized textural features of the randomly selected samples of a book. This task is essential for finding the k_{opt} book content types defined by similar texture indices in the whole book. Finally, we obtain k_{opt} clusters for randomly selected foreground samples of a book, *i.e.* k_{opt} clusters of selected texture vectors computed from a few pages of a book, representing k_{opt} similar content types.

5.3.2.2. Pixel-labeling

This phase deals with labeling clusters or groups of pixels with respect to the results of the pixel-clustering phase. The idea of this task (*cf.* block 1, Figure 5.1) is to assign a label to each cluster of pixels which shares similar textural characteristics to the cluster obtained from the selected foreground samples of the book pages (*cf.* block 2, Figure 5.1).

Journet *et al.* [1] performed the clustering stage using CLARA which is suitable for large scale databases, in the extracted texture features computed from six pages of the same book. Then, if two pixels of two different HDIs have the same cluster label, they belonged to the same class. However, this technique is characterized by a long processing time and memory complexity.

In this work, an unsupervised task is integrated that automatically labels content pixels with the same cluster identifier as the book content. For the same book, each cluster (represented by a given color) represents a similar or homogeneous region. Thus, by applying the HAC algorithm, we group foreground pixels having similar page content type (*i.e.* sharing similar texture indices) defined by similar texture indices. Then, we perform the nearest neighbor search algorithm (NNS) [379] in order to assign the same label to each similar cluster extracted from the digitized book. The NNS is used between each texture feature vector with each digitized page of the same book and the k_{opt} clusters of the selected samples of a book in order to find the texture feature vector closest to the cluster of the selected foreground samples of a book, *i.e.* by selecting the minimum distance.

The NNS is used with the Mahalanobis distance to assign the same label for each similar cluster extracted from a digitized book [380]. The Mahalanobis distance (MD) can be defined as a measure of dissimilarity between two vectors. The MD takes into account dataset correlations and is particularly suited to arbitrarily shaped clusters. The MD is computed of each texture feature vector for each digitized page of the same book from the reference sample defined by each cluster of the selected foreground samples of a book. The Mahalanobis distance $MD(x, y)$ of two multivariate vectors $x = (x_1, x_2, \ldots, x_{Nf})^T$ and $y = (y_1, y_2, \ldots, y_{Nf})^T$ of the same distribution N^f with the covariance matrix S, is defined as:

$$MD(x, y) = \sqrt{\|(x - y)^T S^{-1}(x - y)\|} \tag{5.3}$$

Since the clustering and labeling phases of the proposed texture-based pixel-labeling framework of DHB content have been performed, the foreground pixel in the digitized book are characterized (*cf.* block 3, Figure 5.1).

5.4. Experiments and results

The following is a set of experiments on a large variety of DHBs and HDIs which is detailed to evaluate the performance of the proposed framework and validate our choice of the used techniques on each step of the proposed texture-based pixel-labeling framework of DHB content. A second experimental corpus of HDIs is selected to assess the different phases of the proposed pixel-labeling framework (*cf.* Section 5.4.1). We have evaluated both qualitatively and quantitatively the effectiveness of the extracted texture-based feature sets in Chapter 4. We have proved that the auto-correlation approach is an effective and efficient texture-based one, particularly for HDIs containing graphics and text (*cf.* Section 4.5.1.4). In addition, we have noted that the Gabor-based approach performs considerably better in segmenting HDIs containing only textual regions with distinct fonts (*cf.* Chapter 4 and particularly Section 4.5.1.4). As a consequence, the auto-correlation and Gabor features are chosen to be assessed on the proposed texture-based pixel-labeling framework of DHB content in this section. Several clustering accuracy metrics and classification accuracy rates are computed to evaluate the performance of the proposed framework using the auto-correlation and Gabor features.

Since, the auto-correlation approach has shown, faster than the Gabor one (*cf.* Section 4.5.1.1), the different phases of the proposed pixel-labeling framework of DHB content (*cf.* blocks 1 and 2, Figure 5.1) have been evaluated using the auto-correlation features, in order to evaluate the robustness of the proposed framework and provide additional insights into its classification accuracy (*cf.* Section 5.4.2). Moreover, the pixel-labeling results of the proposed framework have been assessed using the Gabor features in Section 5.4.3.

5.4.1. Experimental protocol

The experimental corpus which is called *"DIGIDOC-Framework dataset"*, is composed of 316 pages which are selected from 13 books in two categories: 7 printed monographs and 6 manuscripts. The *"DIGIDOC-Framework dataset"* encompasses five centuries of French history (1200-1700). For each category, three kinds of content are selected:

- 100 pages containing graphics and a single text font (*"One font and graphics"*):
  - 50 gray-scale manuscript pages (1201-1300)[1]
  - 50 color printed pages (1473)[2]
- 106 pages containing graphics and text with two different fonts (*"Two fonts and graphics"*):

[1] `http://gallica.bnf.fr/ark:/12148/btv1b90075392/f1.planchecontact.r=.langFR`
[2] `http://gallica.bnf.fr/ark:/12148/bpt6k8400870/f11.planchecontact.r=.langFR`

  - 50 manuscript pages: 25 gray-scale pages (1601-1700)[3], 15 gray-scale pages (1301-1500)[4] and 10 gray-scale pages (1201-1300)[5]
  - 56 printed pages: 4 gray-scale pages (1598)[6], 9 gray-scale pages (1594)[7], 4 gray-scale pages (1591)[8], 6 gray-scale pages (1599)[9], 5 gray-scale pages (1594)[10] and 28 gray-scale pages (1711)[11]

- 110 pages containing only two fonts (*"Only two fonts"*):
  - 50 gray-scale manuscript pages (1601-1700)[12]
  - 60 printed pages: 30 gray-scale pages (1594)[13], 10 gray-scale pages (1591)[14] and 20 gray-scale pages (1599)[15]

Some examples of the *"DIGIDOC-Framework dataset"* are shown in Figure 5.3 used to evaluate the performance of the proposed framework and validate our choice of the used techniques on each step of the proposed framework. The *"DIGIDOC-Framework dataset"* is composed of gray-scale/color HDIs which are digitized at 300/400 dpi and saved in the TIFF format which provides a high resolution of digitized images.

5.4.2. Evaluation and results using the auto-correlation features

Historically, we have started exploring the auto-correlation features in this work. As a matter of fact, in this section the different phases of the proposed pixel-labeling framework of DHB content (*cf.* blocks 1 and 2, Figure 5.1) have been evaluated using the auto-correlation features, in order to evaluate the robustness of the proposed framework and provide additional insights into its classification accuracy. The evaluation of the different phases of the proposed framework (*cf.* blocks 1 and 2, Figure 5.1, Sections 5.3.1 and 5.3.2) is described in this section: the estimation of the number of book content types (*cf.* Section 5.3.1), pixel-clustering (*cf.* Section 5.3.2.1) and pixel-labeling (*cf.* Section 5.3.2.2). Several clustering accuracy metrics and classification accuracy rates are computed to assess the different phases of the framework and subsequently to evaluate its performance.

5.4.2.1. Evaluation of the estimation of the number of book content types

The estimated number of book content types is obtained using the merge CCl method with the extracted features of a number of selected foreground pixels chosen randomly from the pages of a book (*cf.* block 2, Figure 5.2, Section 5.3.1).

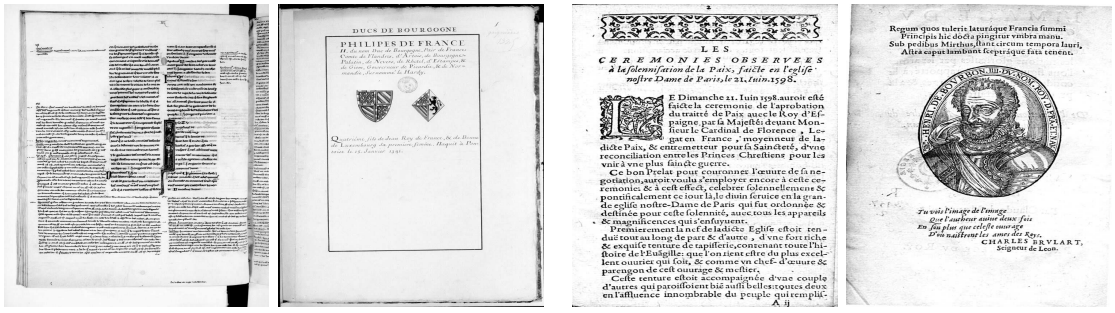
An example of Δk is shown in Figure 5.4(b). In this experiment, k_{opt} is equal to 3 and is estimated from the peak in Δk values of the merge curve.

Table 5.1 shows 10 examples of the estimation of the number of book content types. These examples illustrate 10 estimations computed from 2 different books containing graphics and single text font using CCl and different clustering techniques. For each set of 1000 randomly selected foreground pixels from 10 pages also selected randomly from the same book, we compare the

[3] http://gallica.bnf.fr/ark:/12148/btv1b9058103r/f1.planchecontact.r=.langFR
[4] http://gallica.bnf.fr/ark:/12148/btv1b9060828s/f1.planchecontact.r=.langFR
[5] http://gallica.bnf.fr/ark:/12148/btv1b90620381/f1.planchecontact.r=.langFR
[6] http://gallica.bnf.fr/ark:/12148/bpt6k1315662/f3.planchecontact.r=.langFR
[7] http://gallica.bnf.fr/ark:/12148/bpt6k1316726/f5.planchecontact.r=.langFR
[8] http://gallica.bnf.fr/ark:/12148/bpt6k132056f/f3.planchecontact.r=.langFR
[9] http://gallica.bnf.fr/ark:/12148/bpt6k132093p/f5.planchecontact.r=.langFR
[10] http://gallica.bnf.fr/ark:/12148/bpt6k1347518/f1.planchecontact.r=.langFR
[11] http://gallica.bnf.fr/ark:/12148/bpt6k840383d/f1.planchecontact.r=.langFR
[12] http://gallica.bnf.fr/ark:/12148/btv1b9058220m/f1.planchecontact.r=.langFR
[13] http://gallica.bnf.fr/ark:/12148/bpt6k1316726/f5.planchecontact.r=.langFR
[14] http://gallica.bnf.fr/ark:/12148/bpt6k132056f/f3.planchecontact.r=.langFR
[15] http://gallica.bnf.fr/ark:/12148/bpt6k132093p/f5.planchecontact.r=.langFR

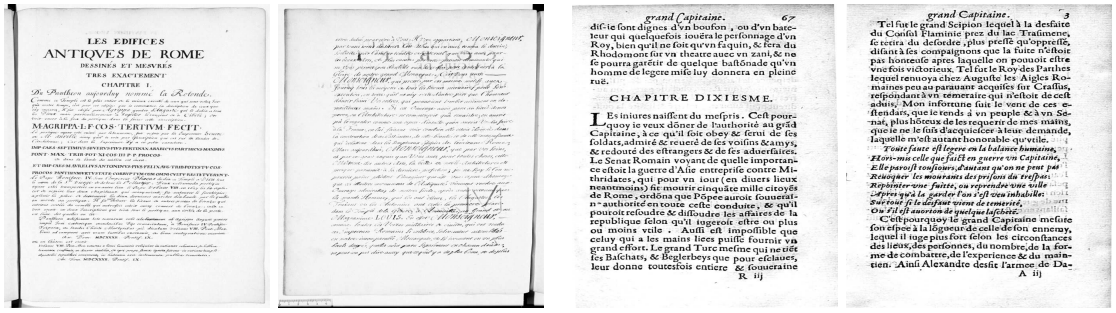
(a) Manuscript-One font and graphics

(b) Printed-One font and graphics



(c) Manuscript-Two fonts and graphics

(d) Printed-Two fonts and graphics



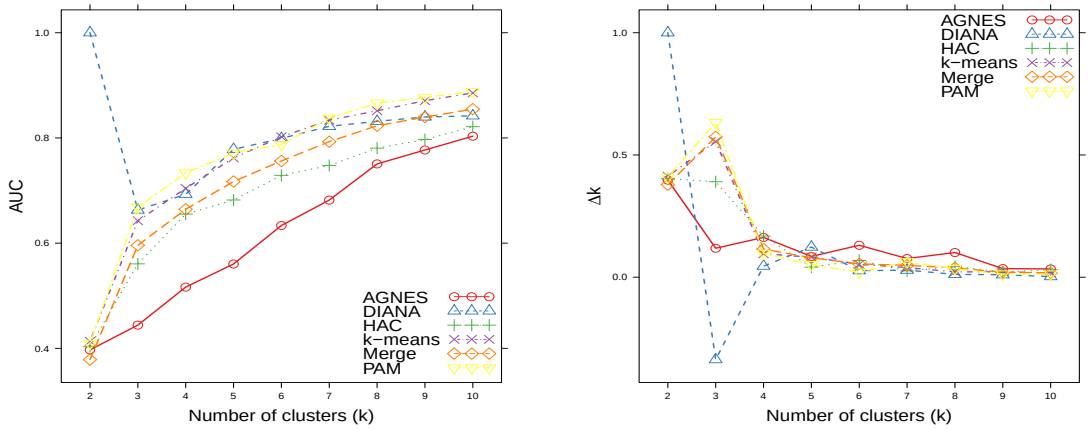
(e) Manuscript-Only two fonts

(f) Printed-Only two fonts

Figure 5.3.: Examples of HDIs from the *"DIGIDOC-Framework dataset"* for the evaluation of the proposed pixel-labeling framework of DHB content.

estimated number of book content types using five clustering methods and the merge CCl technique with the number of clusters set in the defined ground-truth. For most of these estimations, carried out using the two partitioning clustering methods (PAM and k-means), the estimated number of clusters is similar to that set in the defined ground-truth. However, there is a slight variability in the number of clusters estimated by the three hierarchical clustering methods: AGNES, DIANA and HAC. This may be explained by the presence of noise in the analyzed HDIs which can have an impact on the estimated number of book content types. Although a slight variability in the estimated number of book content types is observed when the merge CCl technique is used, the results are relatively consistent since noise and degradation information are taken into consideration. Noise and degradation information can be considered as particular textured areas, *i.e.* characterized by different texture features vectors and which can subsequently constitute a separate cluster. In addition, the results estimate using the PAM method are relatively similar to those obtained with the merge CCl technique and also the number of clusters defined in the ground-truth. Thus, this confirms our hypothesis that the partitioning clustering methods are relatively robust and justifies the higher weight of $\frac{1}{4}$ assigned to each partitioning clustering algorithm (PAM and k-means).

1. 1000 ***vs.*** 2000 ***pixels are used in the CCl technique***



(a) (b)

Figure 5.4.: Illustration of the estimation of the number of book content types using the CCl method. Figure (a) illustrates a plot of AUC for the consensus matrix for each clustering experiment against number of clusters k. Figure (b) depicts a plot of Δk changes in AUC for the consensus matrix for each clustering experiment against number of clusters k. Using the three hierarchical clustering methods: AGNES, DIANA and HAC give an estimate of 2 as the optimal number of clusters, while an estimate of 3 is obtained with k-means, PAM and Merge (the orange curve representing the merge CCl peaked at $k = 3$).

To analyze the robustness of the estimation obtained with the merge CCl technique in the proposed framework, the number of selected foreground pixels introduced as input is varied. Table 5.1 shows the evaluation of the estimation of the number of homogeneous or similar content regions for an analysis 10 sets of 2000 randomly selected pixels from the same 2 different DHBs. The results are in agreement with the number of clusters defined in the ground-truth. We conclude that the effectiveness of the merge CCl technique depends on the number of observations. The higher the number of observations, *i.e.* the number of randomly selected foreground pixels, the better the estimation results. Nevertheless, when numerical complexity is taken into account, a high number of observations requires 16 times the computation time to obtain half of the observations.

2. ***Merge CCl technique vs. internal clustering evaluation measures***
The changes in various clustering evaluation indices are then analyzed for a range of numbers of clusters, and computed from the analysis of the extracted texture features of the selected pixels chosen randomly from pages of a book. For each category of book (*"Printed-One font and graphics"*, *"Manuscript-One font and graphics"*, *"Printed-Two fonts and graphics"*, *"Manuscript-Two fonts and graphics"*, *"Printed-Only two fonts"* and *"Manuscript-Only two fonts"*), the adapted HAC algorithm with the Ward criterion is used for a range of number of clusters after choosing a random selection of a number of foreground pixels from pages of a single book.

   A plot is made of the changes in different clustering evaluation measures over a range of numbers of clusters $[2, 10]$. The results of changes in 9 different internal clustering evaluation measures (silhouette width index, Davies-Bouldin index, Dunn index, Calinski-Harabasz index, Hartigan index, Krzanowski-Lai index, weighted inter-intra measure and homogeneity and separation indices), over a range of numbers of clusters $[2, 10]$ are presented in Figure 5.5. For example, the maximum value of the silhouette width index, the Dunn index, the

Table 5.1.: Examples of the estimation of the number of book content types by analyzing 10 sets of 1000 and 2000 randomly selected pixels from 2 different DHBs containing graphics and single text font with the merge CCl and different clustering techniques.

		Set number											Set number									
		Book 1					Book 2						Book 1					Book 2				
		1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}		1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
AGNES	1000 pixels	2	2	2	2	2	6	4	6	4	5	2000 pixels	5	3	2	3	2	5	4	6	4	5
DIANA		2	4	2	2	2	3	3	2	2	3		2	2	2	2	2	3	2	3	2	3
HAC		2	3	2	3	2	3	3	3	3	4		2	3	2	2	2	4	2	3	3	4
k-means		2	2	2	2	2	2	2	2	2	3		2	2	2	2	2	2	2	2	2	3
PAM		2	2	2	2	2	2	2	2	2	3		2	2	2	2	2	2	2	2	2	2
Merge		2	2	2	2	2	3	3	3	3	3		2	2	2	2	2	2	2	2	2	2
Ground-truth		2	2	2	2	2	2	2	2	2	2		2	2	2	2	2	2	2	2	2	2

Krzanowski-Lai index, the Calinski-Harabasz index and the homogeneity and separation indices are taken to indicate the correct number of clusters in the data, whereas the minimum value of Davies-Bouldin index and the weighted inter-intra measure. On the other hand, the criteria used to select the optimal number of clusters for the Hartigan index is the maximum difference between the hierarchy levels of the index. Figure 5.5 shows that selecting the best number of clusters for each category of book is different for each cluster validation measure and the changes in 9 different internal clustering evaluation measures are not stable. For example, the *"Manuscript-Two fonts and graphics"* category (*cf.* pink curve in Figure 5.5) gives 2, 2, 2, 2, 2, 2, 4, 2 and 2 as the optimal number of clusters estimated by the silhouette width index, Davies-Bouldin index, Dunn index, Calinski-Harabasz index, Hartigan index, Krzanowski-Lai index, weighted inter-intra measure, homogeneity indice and separation measure, respectively. On the other hand, for the *"Printed-Only two fonts"* category (*cf.* magenta curve in Figure 5.5), we obtain 3, 2, 2, 2, 2, 2, 4, 2 and 4 as the optimal number of clusters estimated by the silhouette width index, Davies-Bouldin index, Dunn index, Calinski-Harabasz index, Hartigan index, Krzanowski-Lai index, weighted inter-intra measure, homogeneity indice and separation measure, respectively. Unfortunately, there is no agreement between the unsupervised validity indices that provide a satisfactory solution for the estimation of the number of clusters. Thus, we conclude that by varying all combinations of the number of clusters, the changes in various clustering evaluation indices are not consistent. Indeed, a large number of clustering evaluation indices have been proposed in the literature [381]. Therefore, due to the large variety of cluster structures and the specificity of each clustering accuracy measure, there is no clustering evaluation index which provides a satisfactory solution. Kryszczuk and Hurley claimed that no single clustering evaluation measure can always outperform the others for the estimation of the number of clusters [374].

In the following tests, the results of the estimated number of book content types are compared using the merge CCl method and various internal clustering evaluation measures. For each estimation approach the sum of the differences between the number of book content types defined in the ground-truth and the estimated number of book content types is computed in order to quantify the difference between the number of clusters *vs.* classes. The lower the value for the difference between the number of clusters *vs.* classes, the better the results. The

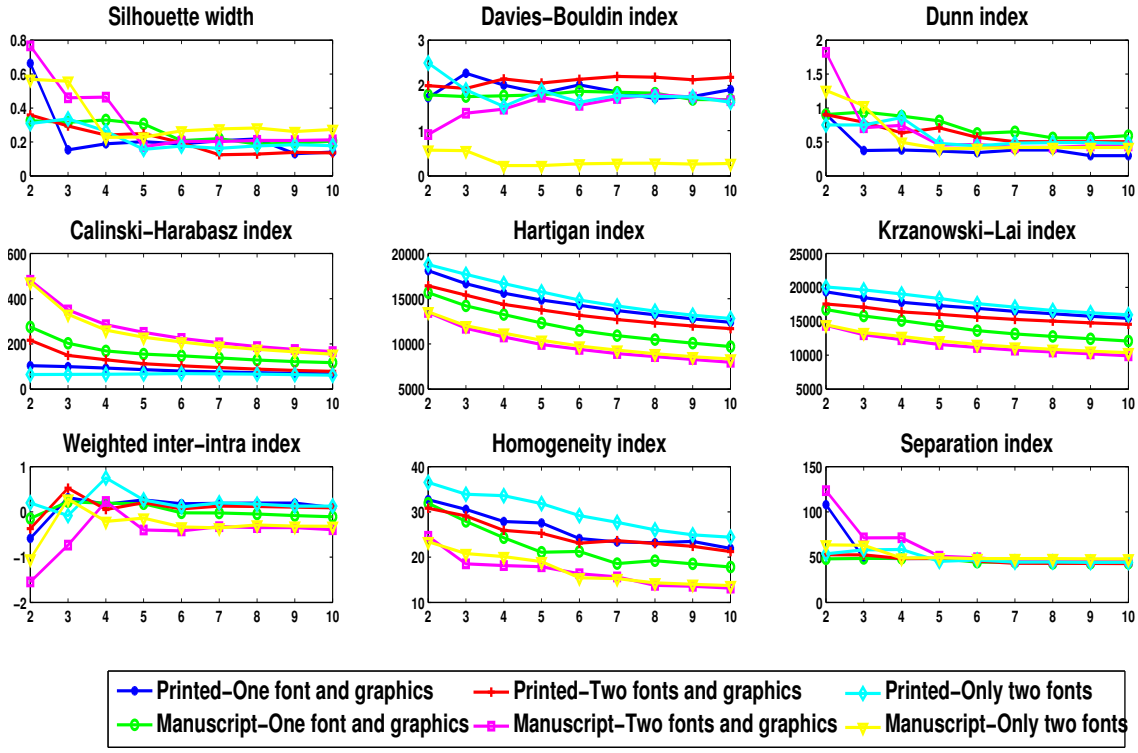


Figure 5.5.: Determination of the optimal number of homogeneous and similar content regions from the results of changes in various internal clustering evaluation indices, over to a range of numbers of clusters and computed from the extracted textural features of the selected foreground pixels chosen randomly from pages of a book. Nine different clustering evaluation measures (silhouette width index, Davies-Bouldin index, Dunn index, Calinski-Harabasz index, Hartigan index, Krzanowski-Lai index, weighted inter-intra measure and homogeneity and separation indices) are made over a range of numbers of clusters $[2, 10]$ and compared for each category of book (*"Printed-One font and graphics"*, *"Manuscript-One font and graphics"*, *"Printed-Two fonts and graphics"*, *"Manuscript-Two fonts and graphics"*, *"Printed-Only two fonts"* and *"Manuscript-Only two fonts"*).

difference between the number of clusters *vs.* classes ($D_k(k_{est}, k_{gt})$) is defined as:

$$D_k(k_{est}, k_{gt}) = \sum_i |k^i_{gt} - k^i_{opt}| \tag{5.4}$$

where k_{est} is the estimated number of clusters, k_{gt} is the number of clusters defined in the ground-truth, and i represents the i^{th} execution of the CCl algorithm on the selected foreground pixels from the digitized book.

Figure 5.6 represents for each category of book the difference between the number of clusters *vs.* classes (D_k) for the CCl method and 21 different internal clustering evaluation measures (Krzanowski-Lai index [382], Hartigan index [383], Calinski-Harabasz index [384], Cubic Clustering Criterion [368], Scott index [385], Marriot index [386], TraceCovW index [387], TraceW index [387], Friedman index [388], Rubin index [389], C-index [390], Davies-Bouldin index [391], silhouette width index [341], Ratkowsky index [392], Ball index [393], PtBiserial index [394], Frey index [395], McClain index [396], Dunn index [397], SDindex [398] and SDbw validity index [399]) (*cf.* Table A.4).

Furthermore, D_k is presented for the CCl technique and the unsupervised evaluation measures for all categories. The lowest differences between the number of clusters *vs.* classes for all

of the *"DIGIDOC-Framework dataset"* are 17, 17 and 16 computed using respectively the CCl, Frey index and McClain index, respectively. Figure 5.6 shows a null value for the difference between the number of clusters *vs.* classes for the Frey index and the McClain index for the printed *"One font and graphics"* document category but this increases for the other categories. The CCl method provides good results for all categories of the *"DIGIDOC-Framework dataset"*, as shown in Figure 5.6. The different experiments detailed in this section prove that the merge CCl technique provides good results by comparing its estimation of the optimal number of clusters with the ground-truth and the internal clustering evaluation indices. However, the CCl method is relatively long so is not particularly effective for a very large dataset.

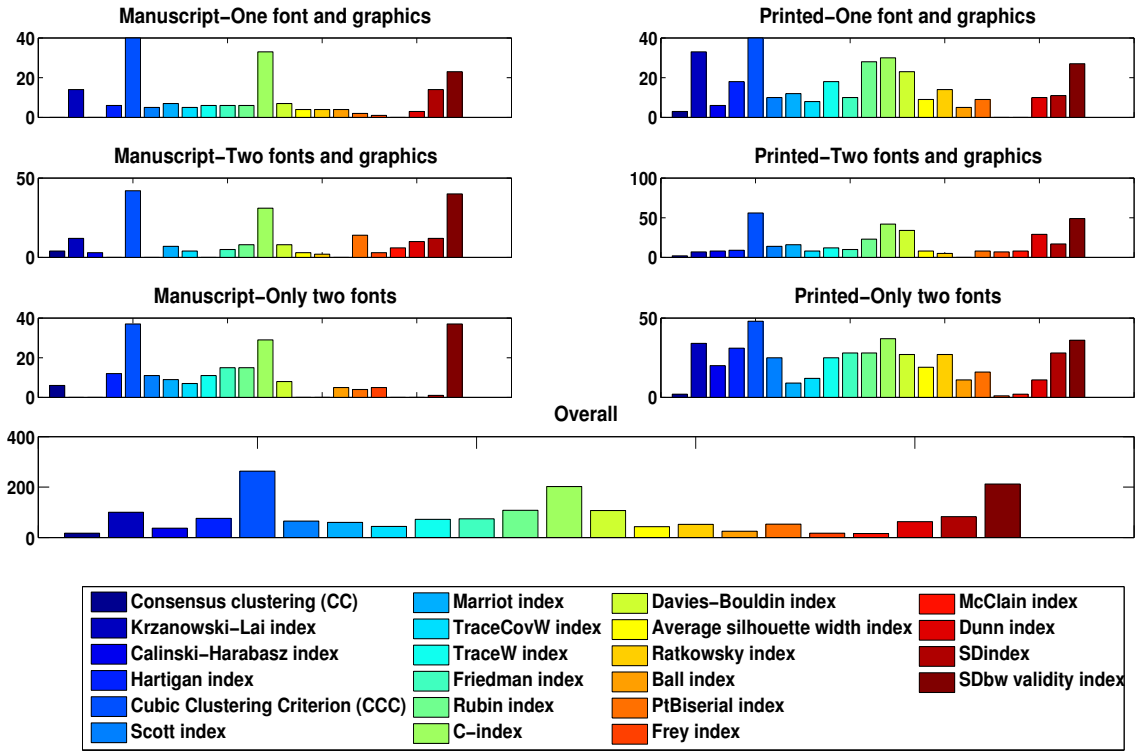


Figure 5.6.: Evaluation of the estimation of the number of book content types by using the CCl method *vs.* various internal clustering evaluation measures: the estimated number of book content types is computed from 21 different internal clustering evaluation measures (Krzanowski-Lai index, Hartigan index, Calinski-Harabasz index, Cubic Clustering Criterion, Scott index, Marriot index, TraceCovW index, TraceW index, Friedman index, Rubin index, C-index, Davies-Bouldin index, silhouette width index, Ratkowsky index, Ball index, PtBiserial index, Frey index, McClain index, Dunn index, SDindex and SDbw validity index) and are compared with the estimated number using the CCl method for each category of DHB (*"Manuscript-One font and graphics"*, *"Printed-One font and graphics"*, *"Manuscript-Two fonts and graphics"*, *"Printed-Two fonts and graphics"*, *"Manuscript-Only two fonts"* and *"Printed-Only two fonts"*). The difference between the number of clusters *vs.* classes for all the *"DIGIDOC-Framework dataset"* is also shown. The lower the difference between the number of clusters *vs.* classes, the better the results.

5.4.2.2. Evaluation of the pixel-clustering phase

The next evaluation step consists in assessing the clustering method, *i.e.* the adapted HAC algorithm with the Ward criterion, used for the pixel-clustering task of the proposed framework (*cf.*

block 1 on Figure 5.1, Section 5.3.2.1).

Indeed, the clustering results obtained after analyzing the extracted texture features are encouraging and indicate many interesting perspectives (*cf.* Section 4.5.1.2). The same proposed clustering approach is therefore used on the *"DIGIDOC-Framework dataset"* of DHB pages. The pixel-clustering results with the HDIs obtained using the HAC algorithm are illustrated in Figure 5.10. The results obtained with the *"DIGIDOC-Framework dataset"* strengthen our previous observations (*cf.* Section 4.5.1.2). Thus, we confirm that the clustering task used with the extracted texture features have a much greater discriminating power for separating text (single font) and graphic regions (*cf.* Figures 5.10(a) and 5.10(b)) than for distinguishing documents containing graphics and two or more text fonts (*cf.* Figure 5.10(c)). The results also confirm that it is more difficult to separate two text fonts (*cf.* Figure 5.10(e)).

5.4.2.3. Evaluation of the pixel-labeling phase

Pixel-labeling (*cf.* block 1 on Figure 5.1, Section 5.3.2.2) is used to determine and assign the same cluster identifier to each similar cluster extracted from the digitized book. This step of the framework uses the NNS technique. This technique is used between each texture feature vector of each digitized page of the same book and the k_{opt} clusters of the selected foreground samples of a book in order to find the closest texture feature vector to the cluster of the selected foreground samples of a book, *i.e.* by selecting the minimum Mahalanobis distance (MD). In order to validate this task, the MD is compared with the Euclidean distance (ED) [400] when the NNS technique is used.

The results of this pixel-labeling step using the ED and MD are illustrated in Figures 5.11 and 5.12, respectively. The success of the pixel-labeling framework is demonstrated by visual inspection of the segmented documents (*cf.* Figures 5.11 and 5.12). The proposed framework gives better results with the MD-based approach (*cf.* Figures 5.12) and finds homogeneous regions in the content of digitized ancient books, *i.e.* for example on Figure 5.12(a) the graphic regions (green) and textual regions (blue) are similarly labeled in two different pages of the same book. It is clear from the four figures 5.12(b), 5.12(a), 5.12(d) and 5.12(c) that the HDIs are segmented into graphic regions which correspond to an ornament and a drop cap and textual regions. For the printed *"Two fonts and graphics"* document category in Figure 5.12(d), the proposed approach distinguishes two different fonts, the normal (blue) and uppercase (green) fonts. On the other hand, Figure 5.12(e) shows that for the manuscript *"Only two fonts"* document category, the proposed approach discriminates between the noise on the HDI borders and the textual regions and separates textual regions with different sizes and fonts, italic and uppercase. However, Figure 5.12(f) suggests that for the printed *"Only two fonts"* document category, the proposed approach can not discriminate between the normal and uppercase fonts when the MD is used.

1. ***Purity per block metric (PPB)***

   Since our goal in this chapter is to find homogeneous or similar content regions defined by similar textural indices (*i.e.* we are not interested in an accurate pixel-based segmentation), the purity per block metric (PPB) (*cf.* equation 4.3), is computed in this section to validate and evaluate a set of experiments.

   - ***Euclidean distance (ED) vs. Mahalanobis distance (MD) is used in the NNS technique***

   The results of PPB (*cf.* equation 4.3) are presented in Table 5.2 when the ED and the MD are used in the pixel-labeling task (*cf.* block 1 on Figure 5.1, Section 5.3.2.2) and after setting the number of randomly selected pixels from 10 pages selected randomly from the same book to 1000 (*cf.* block 2 on Figure 5.1, Section 5.3.1). We obtain $87\% \pm 0.04$ and $85\% \pm 0.04$ mean PPB when using the ED and MD distances are used in the pixel-labeling task, respectively. The overall results are quite satisfying, especially for the manuscript category which, contains

textual (one or two fonts) and non-textual regions. The mean PPB when using the ED and MD are 91% and 92%, respectively, for the manuscript *"One font and graphics"* document category. It can be assumed that manuscripts contain graphic regions that are more compact and homogeneous than printed documents. 94% and 90% mean PPB are obtained with the ED and the MD, respectively, for the category of printed documents containing only two text fonts. The high mean PPB obtained for the printed *"Only two fonts"* document category (*cf.* Figure 5.12(f)) does not signify a good segmentation of the document content according to different types of content regions, *i.e.* different text font. However, it does give an idea about the level of region homogeneity. Hence, further analysis is required with numerous clustering and classification evaluation metrics.

By comparing the average values of PPB for different document categories, a higher mean PPB is obtained for pages containing graphics and single text font when using the ED and the MD. This suggests that the extracted texture features can distinguish textual and graphical regions. When using the ED, an overall higher value of PPB is observed for printed documents than when using the MD for manuscripts. However, a higher standard deviation of PPB is observed when using the ED compared to the MD. Thus, the comparative analysis of the two distances (ED and MD) demonstrates that the synthesis of PPB values for different document categories gives different results depending on the context (text *vs.* graphics, manuscript *vs.* printed document). The analysis of PPB is not sufficient and an additional comparative synthesis is needed, with numerous clustering and classification evaluation accuracies. Nevertheless, if we compare the visual results when using the ED (*cf.* Figure 5.11) and the MD (*cf.* Figure 5.12), it is clear that the best results are obtained with the MD.

Table 5.2.: Purity per block metric (PPB) results of the proposed pixel-labeling framework for DHB content. μ and σ are the mean and standard deviation values of PPB, respectively. The higher the mean values, the better the results. *"*"*: NNS with the Euclidean distance (ED); *"**"*: NNS with the Mahalanobis distance (MD).

Document category	**Document content**	$\mu^{*}(PPB)$	$\sigma^{*}(PPB)$	$\mu^{**}(PPB)$	$\sigma^{**}(PPB)$
Manuscript	One font and graphics	0.91	0.05	0.92	0.01
	Two fonts and graphics	0.90	0.07	0.88	0.04
	Only two fonts	0.81	0.09	0.85	0.05
	Overall	**0.87**	**0.07**	**0.88**	**0.03**
Printed	One font and graphics	0.88	0.14	0.77	0.05
	Two fonts and graphics	0.82	0.07	0.76	0.04
	Only two fonts	0.94	0.05	0.90	0.08
	Overall	**0.88**	**0.09**	**0.81**	**0.03**
Overall		**0.87**	**0.08**	**0.85**	**0.04**

- ***1000 vs. 2000 pixels are used in the CCl technique***

Since the use of the Mahalanobis distance (MD) in the proposed framework is validated using the NNS technique in the pixel-labeling task (*cf.* block 1 on Figure 5.1, Section 5.3.2.2), we will assess in this section the pixel-labeling results with a variable number of selected pixels introduced as input in the estimation of the number of book content types and similar content regions (*cf.* block 2 on Figure 5.2, Section 5.3.1). Thus, the difference in PPB accuracy metric is computed for the results of the pixel-labeling task by setting the number

of randomly selected pixels to 1000 and 2000 in the framework phase: the estimation of the number of book content types (*cf.* block 2 on Figure 5.2, Section 5.3.1). On average, a similar level of pixel-labeling is obtained when 1000 *vs.* 2000 pixels are used in the CCl technique (*cf.* Table 5.3 and Figure 5.13) with an overall difference of 4%. We conclude that the fixed number of randomly selected pixels introduced in the estimation of the number of book content types and similar content regions does not have a negative impact on the results of the pixel-labeling phase of the proposed framework. Thus, we can limit the number of randomly selected pixels for this estimation to 1000.

Table 5.3.: Difference in PPB for pixel-labeling when 1000 *vs.* 2000 pixels are used in the CCl technique.

Document category	**Document content**	***PPB***
Manuscript	One font and graphics	0.01
	Two fonts and graphics	0.06
	Only two fonts	0.12
	Overall	**0.06**
Printed	One font and graphics	0.01
	Two fonts and graphics	0.02
	Only two fonts	0.05
	Overall	**0.03**
Overall		**0.04**

2. ***Other clustering evaluation accuracies***
An additional analysis with different internal and external clustering evaluation indices is needed in order to evaluate the proposed approach, to validate the external evaluation metric, the purity per block measure (PPB) (*cf.* equation 4.3) and the choice of computed distance. In this context, 12 clustering evaluation indices are computed: five internal (Davies-Bouldin index [391], Dunn index [397], Calinski-Harabasz index [384], Hartigan index [383] and Krzanowski-Lai index [382]) and seven external (Rand index [401], adjusted Rand index [402], mutual information measure [403], adjusted mutual information measure [404], J [342], FM [405] and PPB (*cf.* equation 4.3)). The higher the mean values, the better the results (except the Davies-Bouldin index, where lower mean values are better). Numerous clustering evaluation indices are computed using the results of the pixel-labeling phase of the proposed framework with the two distances, ED and MD.

Figure 5.7 shows that the best clustering results are obtained with the most computed evaluation accuracy metrics obtained for the manuscript *"One font and graphics"* document category when using the ED and MD. J and FM are congruent when using MD in pixel-labeling for the following categories of book: *"Manuscript-One font and graphics"*, *"Manuscript-Two fonts and graphics"*, *"Printed-Two fonts and graphics"*, *"Manuscript-Only two fonts"*, *"Printed-Only two fonts"* and *"Overall"*. However, the results obtained using the two computed distances vary and this is observed with the other evaluation accuracy metrics. Moreover, the second best result is obtained for the manuscript *"Only two fonts"* document category with the following six accuracy clustering metrics: Davies-Bouldin index, Calinski-Harabasz index, Rand index, J, FM and PPB. This may be explained by the fact that texture features discriminate between the noise in the HDI and the textual regions (*cf.* Figure 5.12(e)). Three measures (adjusted Rand index, mutual information measure and adjusted mutual information measure) give the second best clustering result for the printed

"Two fonts and graphics" document category (*cf.* Figure 5.12(d)). We can confirm that the probability and information theory based accuracies are relatively concordant. The J, FM and rand index show that the lowest values are obtained for the manuscript *"Two fonts and graphics"* document category. On the other hand, the lowest outcomes for both the PPB and Davies-Bouldin indices are observed in the printed *"Two fonts and graphics"* document category. The three probability and information theory based accuracies (adjusted Rand index, mutual information measure and adjusted mutual information measure) and the Calinski-Harabasz index, the Hartigan index and the Krzanowski-Lai index, suggest that the printed *"Only two fonts"* document category has the lowest outcomes (*cf.* Figure 5.12(f)). It should remembered that the various supervised and unsupervised measures do not evaluate and assess the same aspects. We conclude that the results of the PPB values are relatively similar to those of the various computed internal and external clustering evaluation indices. The best clustering results are obtained for the manuscript *"One font and graphics"* document category with the different clustering evaluation metrics. This strengthens our previous results and confirms our assumption that the texture attributes generally provide the main orientation of a texture (horizontal orientation for textual regions, although there are many orientations that are present to different extents in graphic blocks). The slight variability in the ranking of clustering performance using numerous internal and external clustering measures together with ED or MD can be explained by the specificity of each clustering accuracy measure. For instance, the information and probabilistic theoretical measures compare the distribution of samples in the clustering result and ground-truth by computing the variation in mutual information.

3. ***Classification accuracy metrics***

To ensure that each pixel is classified correctly and to provide additional insight into the classification accuracy, the confusion matrix for each document category is used to compute five measures of classification accuracy: purity (PT), entropy (E), precision (P), recall (R) and classification accuracy rate (CA) (*cf.* Appendix A and particularly Section A.2.2). To evaluate documents containing two fonts and graphics (*cf.* Figures 5.12(c) and 5.12(d)) using clustering and classification accuracy metrics, all fonts in the text have the same label in the ground-truth. Tables 5.4 and Figure 5.8 shows the results of these five classification accuracy measures obtained using the ED and MD in the pixel-labeling task.

We conclude from Tables 5.4 and Figure 5.8 that the results obtained by the numerous computed clustering evaluation measures are coherent with the different classification accuracy results since a considerable improvement in pixel-labeling is obtained when using the MD, with overall gains of 6%(P), 10%(R) and 14%(CA). However, we observe slight drops in the average of 1%(PT) and 0.5%(E). This can be explained by the particular inconsistency of the two classification accuracy metrics which can not indicate precisely the level of accuracy of the results. The best classification for manuscripts containing one font and graphics, especially when using the MD in pixel-labeling (*i.e.* we note gains of 1%(PT), 0.5%(E), 21%(P), 20%(R) and 20%(CA)). However, relatively low classification accuracy metrics (58%(P), 51%(R) and 75%(CA)) are seen for the printed *"One font and graphics"* document category. This low values are unexpected for this category since we have demonstrated a PPB of 77% (*cf.* Table 5.2) without taking into account the spatial relationships. This may raise questions about the defined ground-truth which is to a certain extent subjective. Figure 5.9 indicates the difference between the ground-truth (*cf.* Figure 5.9(e)) and the clustering results (*cf.* Figure 5.9(c,d)). The ground-truth is defined by considering the drop caps as graphic regions while the small letters at the beginning of each text line are considered as text regions (*cf.* Figure 5.9(e)). Nevertheless, the results of pixel-labeling show that the textural characteristics of each small letter at the beginning of each text line is different from the other text

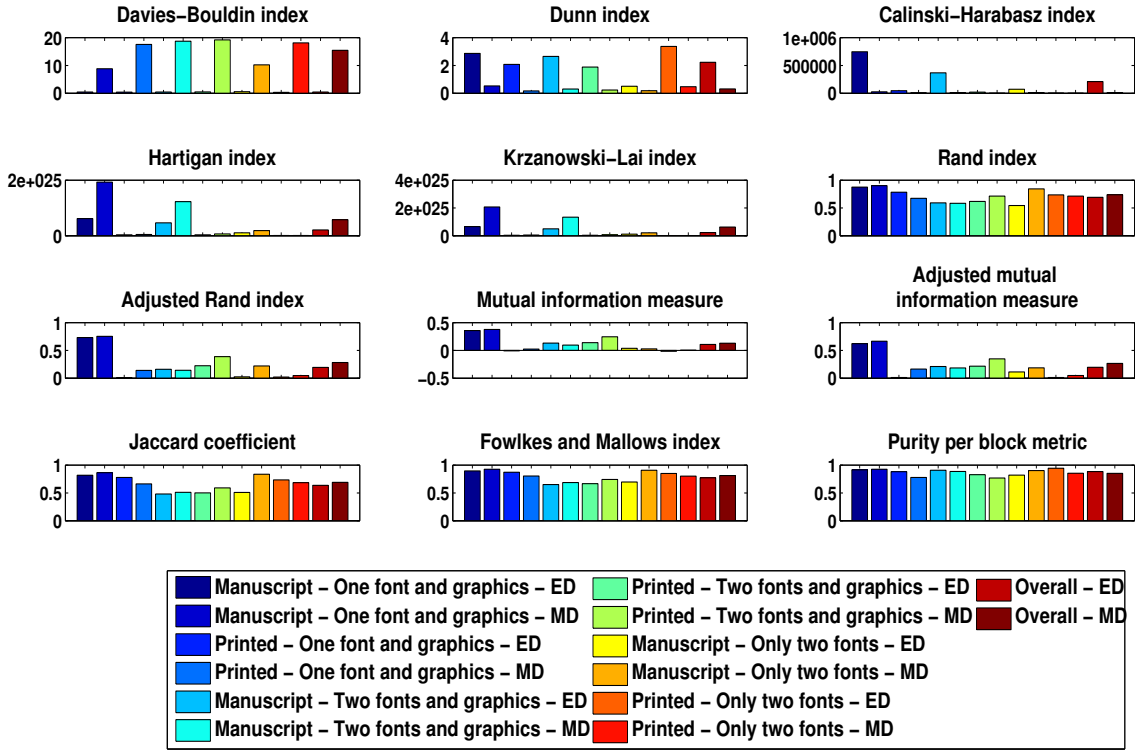


Figure 5.7.: Evaluation of the proposed pixel-labeling framework to DHB content by internal and external clustering accuracy measures performed with the ED and MD in the pixel-labeling task. 12 clustering evaluation indices are used: five internal (Davies-Bouldin index, Dunn index, Calinski-Harabasz index, Hartigan index and Krzanowski-Lai index) and seven external (Rand index, adjusted Rand index, mutual information measure, adjusted mutual information measure, Jaccard coefficient, Fowlkes-Mallows index and purity per block measure). The higher the mean values, the better the results (except the Davies-Bouldin, where lower mean values are better).

content (*cf.* Figure 5.9(c,d)). Thus, to deal with this classic problem, it might be possible to refine the definition of the ground-truth, by taking account of many users' impressions of the ground-truth under consideration. Our previous conclusions on the difficulty of the extracted auto-correlation attributes to separate two or more text fonts (*cf.* Figure 5.12(f)), are demonstrated by computing quantitative clustering accuracy, including external and internal measures. Moreover, calculating the classification accuracy metrics confirms that the extracted auto-correlation indices can not discriminate between two different fonts in particular, italic and normal fonts (*cf.* Figure 5.12(f)). Nevertheless, a slight improvement is observed for classification accuracy measures with manuscripts containing only two fonts characterized by different sizes (*cf.* Figure 5.12(e)). This confirms our assumption that texture features mainly provide the major orientation of the information, *i.e.* the main orientation of the italic font is different from the uppercase one. However, satisfying results are obtained for the printed *"Two fonts and graphics"* documents (*cf.* Figure 5.12(d)). Figure 5.12(d) shows the good results obtained for the segmentation of different kinds of information in the content of printed documents containing graphics (red) and two different fonts: italics (blue) and uppercase (green) fonts. Figure 5.12(c) shows that it is not possible to distinguish two different fonts characterized by different sizes, although the proposed framework separates the graphic (blue), noise (red) and text (green) regions. The high values for mean P, mean R and mean CA for the printed *"Two fonts and graphics"* documents indicate that the proposed pixel-labeling framework tends to mis-classify fewer pixels than for the *"Two fonts and graph-*

ics" manuscripts and indicates that the quality of segmentation and classification depends on the characteristic information content of the analyzed documents. We obtain 73%(P), 72%(R) and 75%(CA) for *"Two fonts and graphics"* manuscripts and 83%(P), 82%(R) and 82%(CA) for printed documents. This confirms our assumption that the manuscripts contain graphic regions that are more compact and homogeneous than the printed documents. The overall results are quite encouraging since we obtain 70%(P), 70%(R) and 79%(CA) for a large variety of ancient books that have many of the particularities of HDIs. These results are based on the extracted texture features, without taking into account the topological or spatial relationships and with no hypothesis concerning the document layout or the typographical parameters of the document. High values are obtained for the classification accuracy metrics (75%(P), 78%(R) and 82%(CA)) with the manuscript category compared to printed documents, *i.e.* difference values of 7%(PT), 0.30%(E), 11%(P), 17% (R) and 6%(CA). This can be justified by the fact that manuscripts are characterized by a particular style which generates structured textural features, *i.e.* manuscripts contain drawing regions that are more compact and homogeneous than the printed documents.

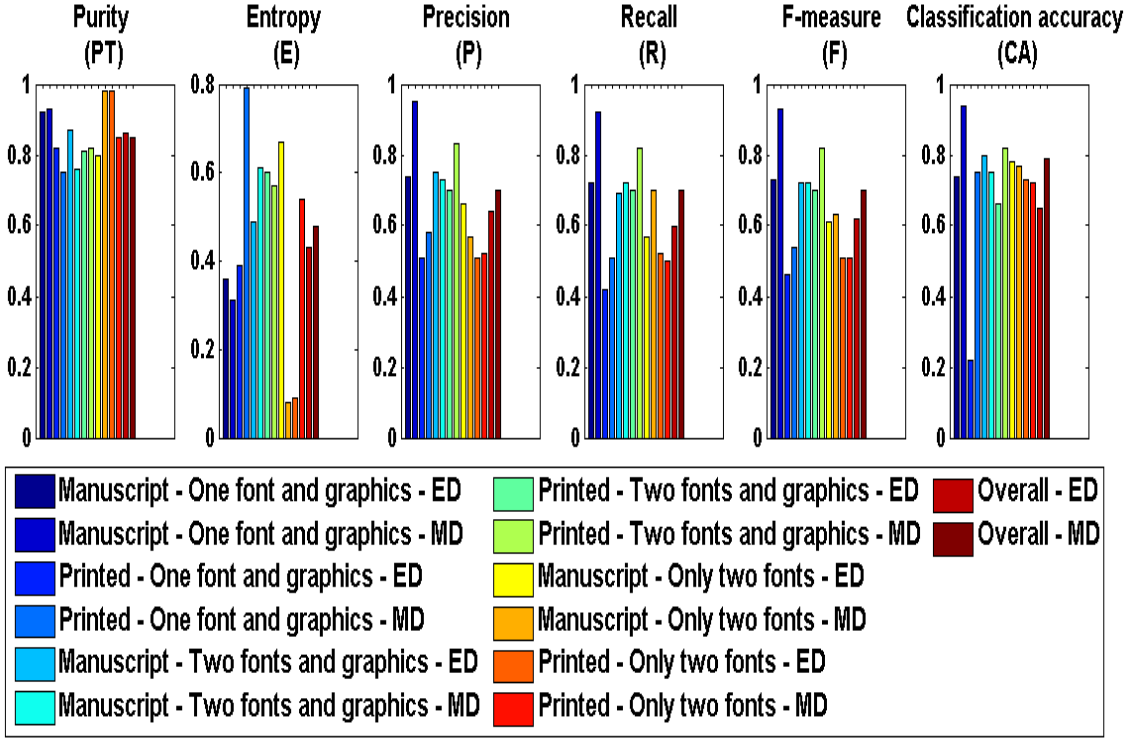


Figure 5.8.: Evaluation of the proposed pixel-labeling framework for DHB content using classification accuracy measures with the ED and MD in the pixel-labeling task. Five classification evaluation indices: PT, E, P, R, F and CA. The higher the mean values are, the better the results (except E, where lower mean values are better).

5.4.2.4. Discussion

In Section 5.4.2, we have demonstrated both qualitatively and quantitatively the effectiveness of the extracted auto-correlation features performed on the proposed texture-based pixel-labeling framework for DHB content. The auto-correlation-based pixel-labeling framework provides a good discrimination of the foreground layers of DHB pages, particularly between text and graphics. This strengthens and confirms our previous results obtained in Chapter 4 (*cf.* Section 4.5.1.4) that the auto-correlation descriptors can distinguish textual from graphic regions of an analyzed HDI.

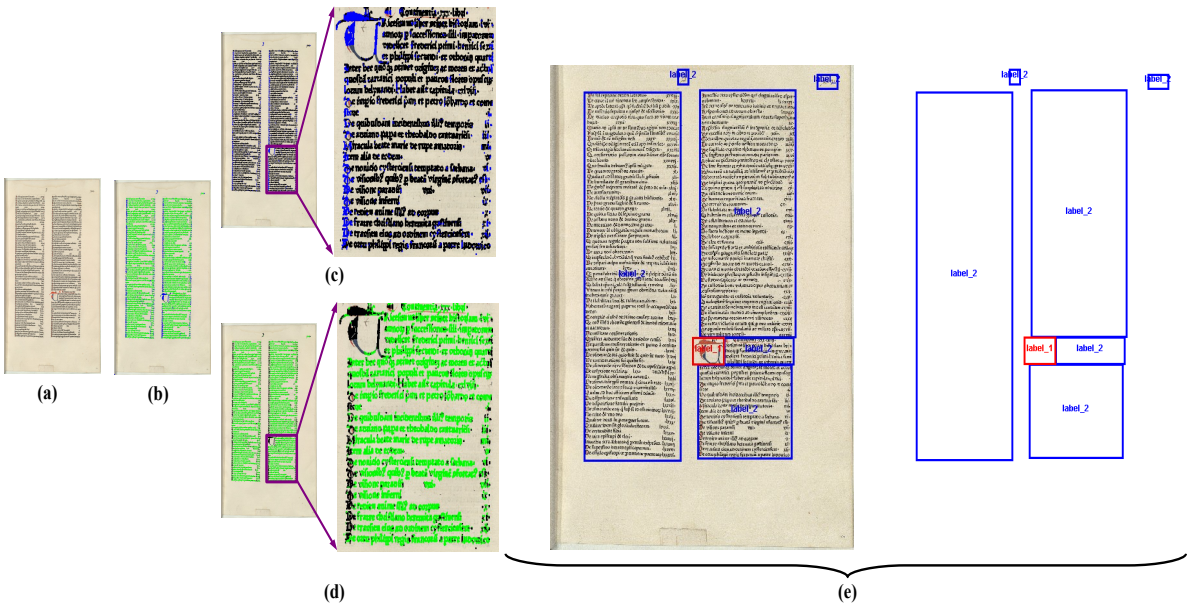


Figure 5.9.: The pixel-labeling result *vs.* ground-truth. Figure (a) illustrates an original gray-scale image. Figure (b) shows the final result of the pixel-labeling task. Figure (c) shows a cluster representing the graphics, while Figure (d) illustrates a cluster representing the text. Figure (e) shows the associated ground-truth.

Concerning the overall results obtained with the proposed auto-correlation-based pixel-labeling framework of DHB content, 85%(PPB), 79%(CA), 70%(P) and 70%(R) are noted with a low processing time and memory complexity. The proposed framework has the advantage that it is performed in the absence of a hypothesis concerning the document layout (physical structure) or the typographical parameters of the document (logical structure). In this framework, the number of book content types is automatically determined using the CCl technique on randomly selected pixels from ten book pages without taking into account the spatial attributes. This approach is based on book page analysis using the CCl and NNS techniques to find similarities between the textural characteristics of the HDI content. In the proposed framework, there is no post-processing of the segmented documents. The results will be improved if a new task is introduced for the use of spatial relationships among the selected pixels.

5.4.3. Evaluation and results using the Gabor features

In order to evaluate the robustness of the proposed texture-based pixel-labeling framework for DHB content on the *"DIGIDOC-Framework dataset"*, we have also assessed it using the Gabor features. This section aims at illustrating the framework genericity with respect to the used texture feature set. As a consequence, our goal is to compare the pixel-labeling results given by the proposed texture-based framework when using two different kinds of texture features, the auto-correlation and Gabor descriptors. We compare the pixel-labeling results obtained by performing the proposed auto-correlation-based pixel-labeling framework for DHB content on the *"DIGIDOC-Framework dataset"* (1000 pixels is introduced into the CCl technique and the MD is used in the pixel-labeling task) with those obtained when replacing the auto-correlation features by the Gabor ones and leaving all other setting parameters unchanged. Qualitative and numerical experiments are given to demonstrate the performance of the Gabor-based pixel-labeling framework for DHB content in Sections 5.4.3.1 and 5.4.3.2, respectively.

5.4.3.1. Qualitative results

The results of the pixel-labeling step by introducing 1000 pixels into the CCl technique and using the MD in the pixel-labeling task are illustrated in Figures 5.14. For the manuscript *"Two fonts and graphics"* document category in Figure 5.14(c), the proposed Gabor-based framework distinguishes graphics (red) and text with two different fonts (*"Font 1"*: text with S_1^f size font (blue) and *"Font 2"*: text with $S_2^f \leq S_1^f$ size font (green)). For the manuscript *"One font and graphics"* document category in Figure 5.14(a), the proposed Gabor-based framework discriminates graphics (green) and one text font (blue). We observe that by comparing the two pixel-labeling results using the auto-correlation (*cf.* Figure 5.12(a)) and Gabor (*cf.* Figure 5.14(a)) features, we obtain more homogeneous regions when using the Gabor feature in the proposed texture-based framework for the manuscript *"One font and graphics"* document category. However, contrarily to the obtained results when using the auto-correlation features (*cf.* Figure 5.12(e)), our Gabor-based framework discriminates only between the noise on the DI borders (blue) and the textual regions (green) and can not separate textual regions with different sizes and fonts, italic and uppercase for the manuscript *"Only two fonts"* category (*cf.* Figure 5.14(e)). This can be justified by the mis-estimation of the number of clusters by means of the CCl technique. On the other side, for the printed *"Only two fonts"* category, the Gabor-based framework (*cf.* Figure 5.14(f)) performs better than the auto-correlation-based framework (*cf.* Figure 5.14(f)) for separating two text fonts, italic (blue and green) and uppercase (red) fonts. Nevertheless, the pixel-labeling results of the Gabor-based framework for the *"One font and graphics"* (*cf.* Figure 5.14(b)) and *"Two fonts and graphics"* (*cf.* Figure 5.14(d)) printed documents are less satisfactory than those obtained with the auto-correlation-based framework (*cf.* Figures 5.12(b) and 5.12(d)). This can be justified by either the limitations of the Gabor features to separate spatially close various kinds of information or the mis-estimation of the number of clusters by means of the CCl technique. Hence, the proposed pixel-labeling framework using the Gabor features is more sensitive to the estimation of the number of clusters by means of the CCl technique than the auto-correlation ones. This can be explained by the range changes in Gabor attribute indices regarding the DHB page layout and/or content, and this can subsequently affect the pixel-labeling results particularly when selecting randomly 1000 foreground pixels from few pages selected randomly from the same book under consideration.

5.4.3.2. Quantitative results

In order to evaluate the robustness of the proposed framework and provide additional insights into its classification accuracy, we have assessed the pixel-labeling results of the proposed Gabor-based framework. Tables 5.5 present the performance of the proposed Gabor-based framework performed with the MD in the pixel-labeling task by computing several clustering and classification accuracy metrics (J, FM, PPB, PT, E, P, R, F and CA). We conclude from Tables 5.5 and Figures 5.14 that the pixel-labeling results obtained by the numerous computed clustering and classification evaluation measures are coherent with the different classification accuracy results when using the Gabor features on the proposed texture-based framework and by introducing 1000 pixels into the CCl technique and using the MD in the pixel-labeling task. A slight decline in the Gabor-based pixel-labeling results is observed compared to those obtained with the auto-correlation features, with overall drops of 1%(E), 1%(P), 5%(R), 2%(CA) and 4%(F). This strengthens our previous observations that the Gabor features fail to separate spatially close various kinds of information due to the range changes in Gabor attribute indices regarding the DHB page layout and/or content (*i.e.* the pixel-labeling results depend on the Gabor attribute indices of the random selection of 1000 foreground pixels from few pages selected randomly from the same book under consideration). Nevertheless, we note a slight improvement of the purity and homogeneity of the regions, with gains of 2%(PPB), 4%(PT). We note that when using the Gabor features on the texture-based pixel-labeling framework leads to more homogeneous and pure regions than when using the auto-correlation features on it. We obtain 94%(PPB), 71%(P), 69%(R), 82%(CA) and 69%(F) for

manuscripts and 84%(PPB), 66%(P), 60%(R), 72%(CA) and 63%(F) for printed documents. Similarly to the pixel-labeling results deduced from the auto-correlation-based framework, we state a considerable outperformance of the pixel-labeling results using the Gabor features for manuscripts compared to the printed documents (*i.e.* gains of 10%(PPB), 5%(P), 9%(R), 10%(CA) and 6%(F)). This confirms our previous assumption deduced when using the auto-correlation features on the proposed framework that the manuscripts contain graphic regions that are more compact and homogeneous than the printed documents. The best pixel-labeling performance is noted for the manuscript *"One font and graphics"* document category (97%(P), 95%(R), 97%(CA) and 96%(F)). On the other side, we note that for the other document categories of the *"DIGIDOC-Framework dataset"* similar performances are observed (about 60%(F)). Therefore, we also confirm that the Gabor-based framework has good ability to separate graphics from text regions due to the particularities of the manuscript style which generates structured textural features.

5.5. Discussion

We note that overall the auto-correlation-based framework performs slightly better than the Gabor one. The pixel-labeling results depend on the estimation of the number of clusters using the CCl technique, the selectivity to the book layout and/or content (*i.e.* layout structure, text *vs.* graphics) and also book characteristics (e.g. manuscript *vs.* printed). Further work needs to be done in combining various kinds of texture descriptors in order to construct an optimal texture-based feature set.

The overall results of the proposed texture-based pixel-labeling framework of DHB content are quite satisfying. However, it is possible to speculate that if we integrate several kinds of post-processing techniques, we will have better results than those reported in this chapter. It is important to be noted that we do not assume knowledge about the font size, scanning resolution, column layout, orientation, *etc.* of the analyzed DI.

Supported by the fact that pages of the same book usually present strong similarities in the organization of the HDI information (*i.e.* layout) and in the graphical and typographical features (*i.e.* content) throughout the DHB pages, our objective is to propose an approach that is used on an entire book instead of processing each page individually, for characterization and categorization of DHB pages and the segmentation and analysis of DHB content. Nevertheless, it is also important to point out that the front page of an ancient book is usually different in design, style, layout and content from other book pages. If an unsupervised clustering is performed over all book pages, how can the typical features appearing on first page only gain enough weight to appear as an independent cluster against the features from all other book pages. In future research, this point will be discussed to present a convenient solution assigning specific processing to the front page of a book.

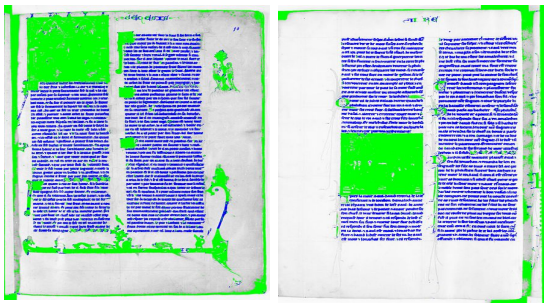
The techniques and parameters used in the proposed texture-based pixel-labeling framework of DHB content, *i.e.* the clustering method, the standard non-parametric binarization method used to retrieve only pixels representing the information in the foreground, the sizes of the sliding windows for the multi-scale approach, the number of selected pixels introduced as input in the estimation of the number of book content types, the distance used in the NNS technique, are selected based on work published in the literature and after performing several experiments to choose the best configuration of the different techniques in the proposed framework. Moreover, a constructive compromise between computation time and pixel-labeling quality is respected.

5.6. Conclusion

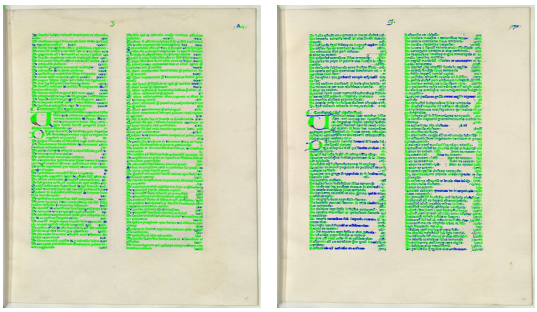
This chapter proposes a generic framework for a texture-based pixel-labeling of DHB content with no hypothesis concerning the document layout or the typographical parameters (*i.e.* typographic/graphical characteristics) of the document. The aim of this framework is to group pixels

having similar DHB page content type within the content of DHBs by extracting and analyzing texture features independently of the layout of the pages. It is therefore applicable to a large variety of books. The proposed framework is based on a feature vector that is composed of texture indices. Texture features are extracted from the different areas of a page and at several resolutions. The robustness of the extracted features is used in a parameter-free unsupervised clustering method which is performed to determine the number of book content types (*i.e.* defined by similar texture indices). Moreover, the number of book content types does not need to be known in advance as it is automatically determined.

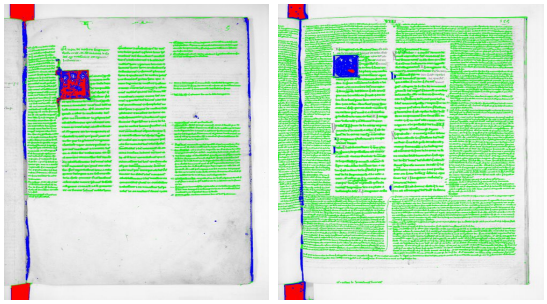
The proposed framework has been evaluated on the *"DIGIDOC-Framework dataset"* which is composed of 316 pages of HDIs. We conclude that texture features provide a good discrimination of the foreground layers of DHB pages, particularly between text and graphics. 85% purity per block accuracy and 79% classification accuracy are obtained for the auto-correlation-based framework, while 89% purity per block accuracy and 77% classification accuracy are noted for the Gabor-based framework.



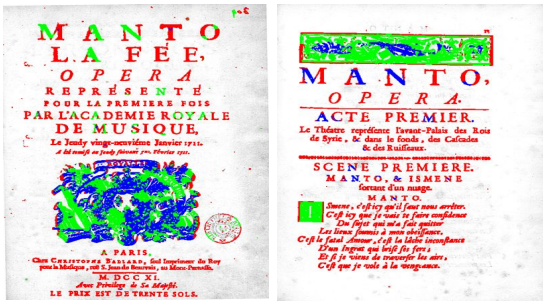
(a) Manuscript-One font and graphics



(b) Printed-One font and graphics

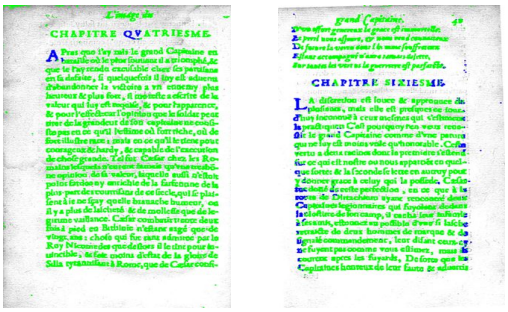


(c) Manuscript-Two fonts and graphics



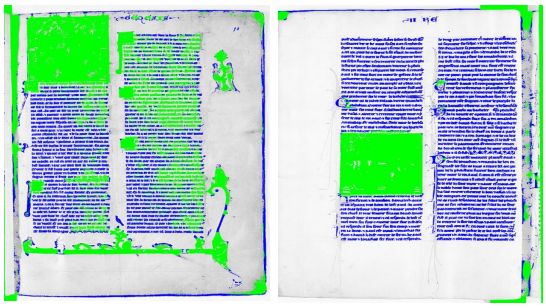
(d) Printed-Two fonts and graphics

(e) Manuscript-Only two fonts

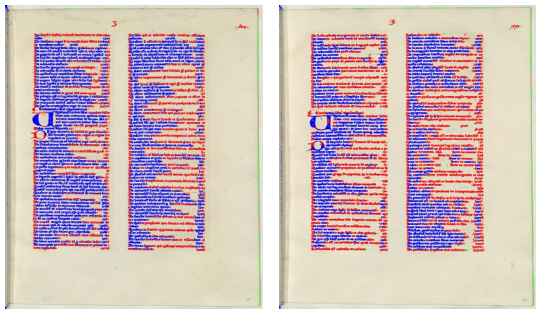


(f) Printed-Only two fonts

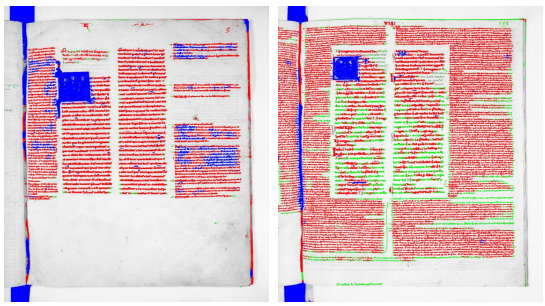
Figure 5.10.: Examples of resulting images of the **pixel-clustering** task used with the **auto-correlation** features on the ***"DIGIDOC-Framework dataset"***. Since the pixel-labeling task is not processed, the colors attributed to text or graphics may differ from one DI to another.



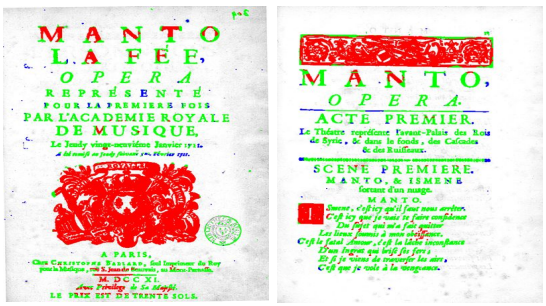
(a) Manuscript-One font and graphics
$PPB = 0.91$ $P = 0.74$ $R = 0.72$ $CA = 0.74$



(b) Printed-One font and graphics
$PPB = 0.88$ $P = 0.51$ $R = 0.42$ $CA = 0.22$

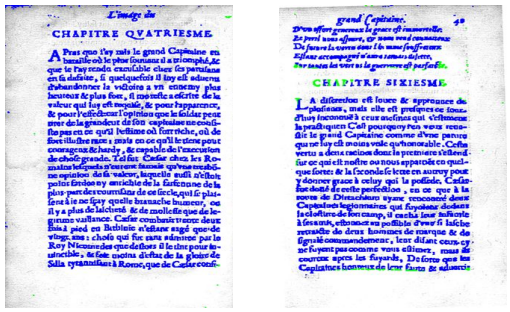


(c) Manuscript-Two fonts and graphics
$PPB = 0.90$ $P = 0.75$ $R = 0.69$ $CA = 0.80$



(d) Printed-Two fonts and graphics
$PPB = 0.82$ $P = 0.70$ $R = 0.70$ $CA = 0.66$

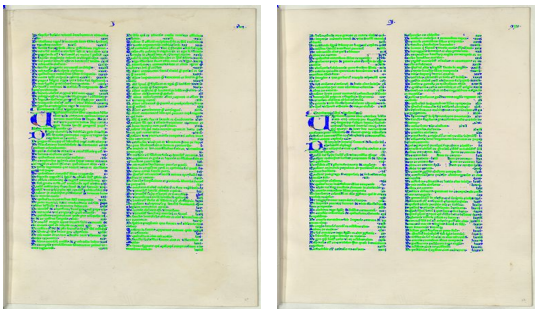
(e) Manuscript-Only two fonts
$PPB = 0.81$ $P = 0.66$ $R = 0.57$ $CA = 0.78$



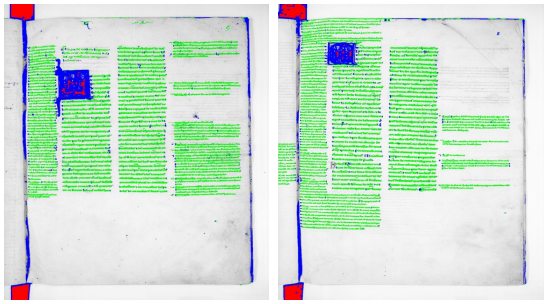
(f) Printed-Only two fonts
$PPB = 0.94$ $P = 0.51$ $R = 0.52$ $CA = 0.73$

Figure 5.11.: Examples of resulting images of the proposed **auto-correlation**-based pixel-labeling framework for DHB content on the ***"DIGIDOC-Framework dataset"***, performed by introducing **1000** pixels into the **CCl** technique and using the **ED** in the **pixel-labeling** task. For the same book, each cluster (represented by a given color) represents a similar or homogeneous region. Since the process is unsupervised, the colors attributed to text or graphics may differ from one book to another.

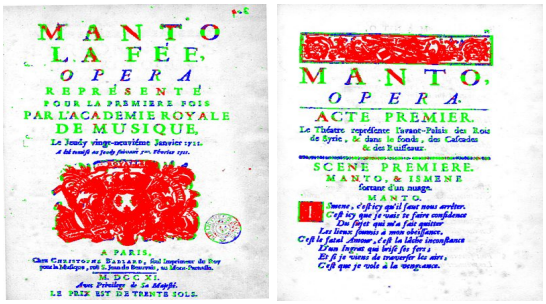
(a) Manuscript-One font and graphics
$PPB = 0.92$ $P = 0.95$ $R = 0.92$ $CA = 0.94$



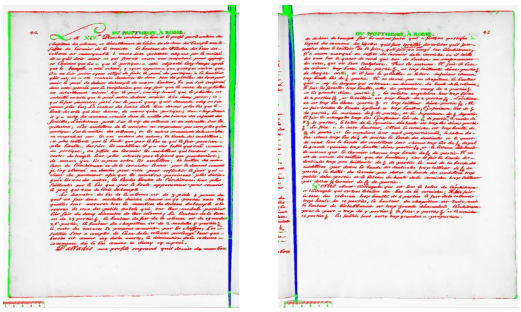
(b) Printed-One font and graphics
$PPB = 0.77$ $P = 0.58$ $R = 0.51$ $CA = 0.75$



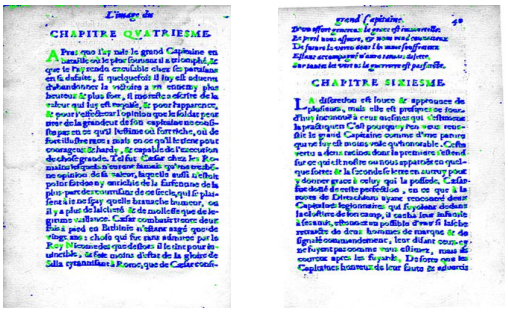
(c) Manuscript-Two fonts and graphics
$PPB = 0.88$ $P = 0.73$ $R = 0.72$ $CA = 0.75$



(d) Printed-Two fonts and graphics
$PPB = 0.76$ $P = 0.83$ $R = 0.82$ $CA = 0.82$

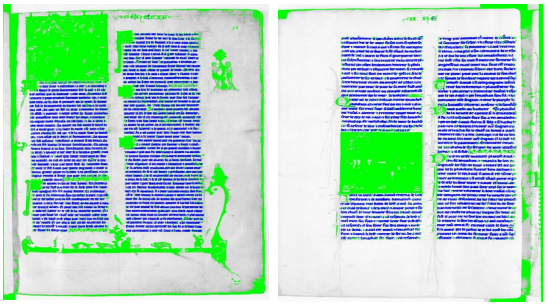


(e) Manuscript-Only two fonts
$PPB = 0.85$ $P = 0.57$ $R = 0.70$ $CA = 0.77$

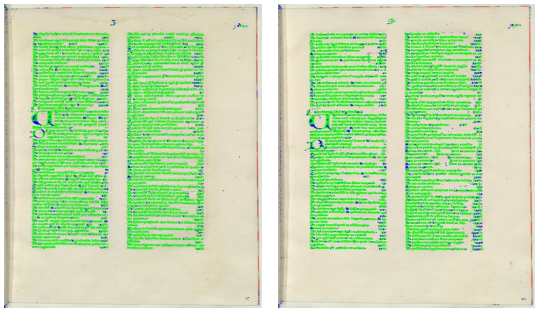


(f) Printed-Only two fonts
$PPB = 0.90$ $P = 0.52$ $R = 0.50$ $CA = 0.72$

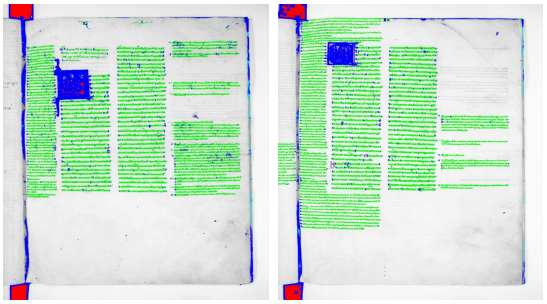
Figure 5.12.: Examples of resulting images of the proposed **auto-correlation**-based pixel-labeling framework for DHB content on the ***"DIGIDOC-Framework dataset"***, performed by introducing **1000** pixels into the **CCl** technique and using the **MD** in the **pixel-labeling** task. For the same book, each cluster (represented by a given color) represents a similar or homogeneous region. Since the process is unsupervised, the colors attributed to text or graphics may differ from one book to another.



(a) Manuscript-One font and graphics
$PPB = 0.91$



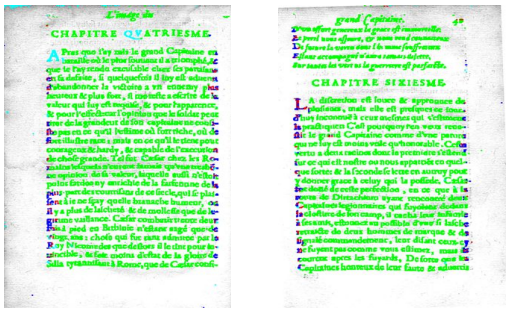
(b) Printed-One font and graphics
$PPB = 0.76$



(c) Manuscript-Two fonts and graphics
$PPB = 0.82$

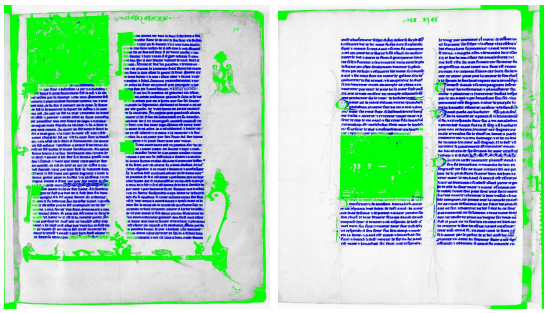
(d) Printed-Two fonts and graphics
$PPB = 0.74$

(e) Manuscript-Only two fonts
$PPB = 0.78$

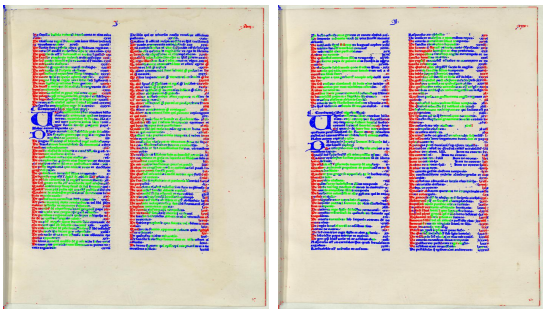


(f) Printed-Only two fonts
$PPB = 0.80$

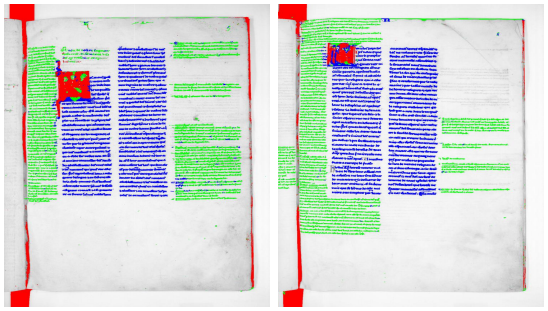
Figure 5.13.: Examples of resulting images of the proposed **auto-correlation**-based pixel-labeling framework for DHB content on the ***"DIGIDOC-Framework dataset"***, performed by introducing **2000** pixels into the **CCl** technique and using the **MD** in the **pixel-labeling** task. For the same book, each cluster (represented by a given color) represents a similar or homogeneous region. Since the process is unsupervised, the colors attributed to text or graphics may differ from one book to another.



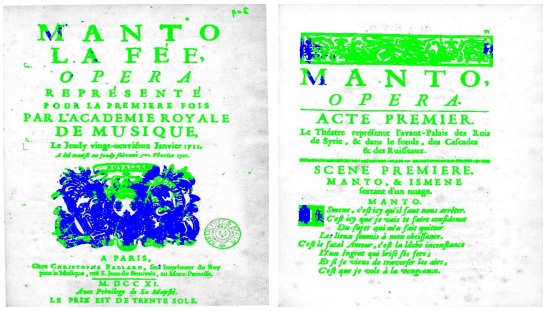
(a) Manuscript-One font and graphics
$PPB = 0.91$ $P = 0.97$ $R = 0.95$ $CA = 0.97$



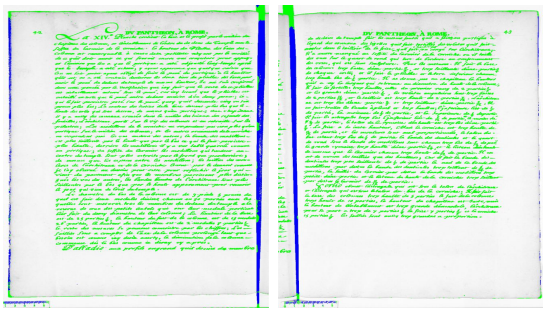
(b) Printed-One font and graphics
$PPB = 0.78$ $P = 0.68$ $R = 0.60$ $CA = 0.82$



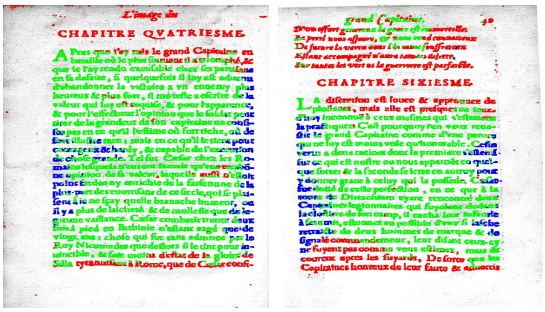
(c) Manuscript-Two fonts and graphics
$PPB = 0.95$ $P = 0.65$ $R = 0.61$ $CA = 0.73$



(d) Printed-Two fonts and graphics
$PPB = 0.92$ $P = 0.62$ $R = 0.64$ $CA = 0.62$



(e) Manuscript-Only two fonts
$PPB = 0.97$ $P = 0.50$ $R = 0.50$ $CA = 0.76$



(f) Printed-Only two fonts
$PPB = 0.83$ $P = 0.67$ $R = 0.57$ $CA = 0.71$

Figure 5.14.: Examples of resulting images of the proposed **Gabor**-based pixel-labeling framework for DHB content on the ***"DIGIDOC-Framework dataset"***, performed by introducing **1000** pixels into the **CCl** technique and using the **MD** in the **pixel-labeling** task. For the same book, each cluster (represented by a given color) represents a similar or homogeneous region. Since the process is unsupervised, the colors attributed to text or graphics may differ from one book to another.

Table 5.4.: Quantitative assessment with numerous **classification accuracy** metrics of the proposed **auto-correlation**-based framework performed by introducing **1000** pixels into the **CCl** technique and using the **ED** and **MD** in the **pixel-labeling** task: purity (PT), entropy (E), precision (P), recall (R), F-measure (F) and classification accuracy (CA). $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results (except E, where lower mean values are better). For documents containing two fonts and graphics (*cf.* Figures 5.12(c) and 5.12(d)), all fonts in the text have the same label in the ground-truth.

	Document category	**Document content**	$\mu(PT)$	$\sigma(PT)$	$\mu(E)$	$\sigma(E)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(CA)$	$\sigma(CA)$	$\mu(F)$	$\sigma(F)$
ED	Manuscript	One font and graphics	0.92	0.02	0.36	0.11	0.74	0.38	0.72	0.38	0.74	0.39	0.73	0.38
		Two fonts and graphics	0.87	0.07	0.49	0.16	0.75	0.19	0.69	0.16	0.80	0.10	0.72	0.17
		Only two fonts	0.80	0.07	0.67	0.16	0.66	0.26	0.57	0.10	0.78	0.09	0.61	0.14
		Overall	**0.86**	**0.05**	**0.51**	**0.14**	**0.72**	**0.28**	**0.66**	**0.21**	**0.77**	**0.19**	**0.67**	**0.24**
	Printed	One font and graphics	0.82	0.24	0.39	0.53	0.51	0.07	0.42	0.06	0.22	0.27	0.46	0.06
		Two fonts and graphics	0.81	0.07	0.60	0.09	0.70	0.17	0.70	0.18	0.66	0.20	0.70	0.17
		Only two fonts	0.98	0.01	0.09	0.05	0.51	0.03	0.52	0.03	0.73	0.18	0.51	0.03
		Overall	**0.87**	**0.11**	**0.36**	**0.22**	**0.57**	**0.09**	**0.55**	**0.09**	**0.54**	**0.22**	**0.56**	**0.09**
	Overall		**0.86**	**0.08**	**0.43**	**0.18**	**0.64**	**0.18**	**0.60**	**0.15**	**0.65**	**0.20**	**0.62**	**0.16**
	Document category	**Document content**	$\mu(PT)$	$\sigma(PT)$	$\mu(E)$	$\sigma(E)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(CA)$	$\sigma(CA)$	$\mu(F)$	$\sigma(F)$
MD	Manuscript	One font and graphics	0.93	0.01	0.31	0.06	0.95	0.02	0.92	0.02	0.94	0.01	0.93	0.02
		Two fonts and graphics	0.76	0.18	0.61	0.34	0.73	0.15	0.72	0.14	0.75	0.17	0.72	0.14
		Only two fonts	0.98	0.01	0.08	0.08	0.57	0.20	0.70	0.30	0.77	0.39	0.63	0.24
		Overall	**0.89**	**0.07**	**0.33**	**0.16**	**0.75**	**0.12**	**0.78**	**0.15**	**0.82**	**0.19**	**0.76**	**0.13**
	Printed	One font and graphics	0.75	0.05	0.79	0.07	0.58	0.24	0.51	0.02	0.75	0.06	0.54	0.04
		Two fonts and graphics	0.82	0.05	0.57	0.13	0.83	0.03	0.82	0.06	0.82	0.05	0.82	0.04
		Only two fonts	0.85	0.07	0.54	0.17	0.52	0.13	0.50	0.13	0.72	0.14	0.51	0.13
		Overall	**0.81**	**0.06**	**0.63**	**0.12**	**0.64**	**0.13**	**0.61**	**0.07**	**0.76**	**0.08**	**0.62**	**0.09**
	Overall		**0.85**	**0.06**	**0.48**	**0.14**	**0.70**	**0.12**	**0.70**	**0.13**	**0.79**	**0.13**	**0.70**	**0.12**

Table 5.5.: Quantitative assessment with numerous **clustering and classification accuracy** metrics of the proposed **Gabor**-based framework performed by introducing the **1000** pixels into the **CCl** technique and using the **MD** in the **pixel-labeling** task: Jaccard coefficient (J), Fowlkes-Mallows index (FM), purity per block measure (PPB), purity (PT), entropy (E), precision (P), recall (R), F-measure (F) and classification accuracy (CA). $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results (except E, where lower mean values are better). For documents containing two fonts and graphics (*cf.* Figures 5.14(c) and 5.14(d)), all fonts in the text have the same label in the ground-truth.

	Document category	Document content	$\mu(J)$	$\sigma(J)$	$\mu(FM)$	$\sigma(FM)$	$\mu(PPB)$	$\sigma(PPB)$
MD	Manuscript	One font and graphics	0.95	0.03	0.94	0.02	0.91	0.05
		Two fonts and graphics	0.63	0.19	0.76	0.15	0.95	0.04
		Only two fonts	0.87	0.09	0.93	0.05	0.97	0.04
		Overall	**0.82**	**0.10**	**0.88**	**0.07**	**0.94**	**0.04**
	Printed	One font and graphics	0.76	0.20	0.86	0.12	0.78	0.11
		Two fonts and graphics	0.58	0.13	0.74	0.09	0.92	0.06
		Only two fonts	0.57	0.14	0.72	0.12	0.83	0.07
		Overall	**0.64**	**0.16**	**0.77**	**0.11**	**0.84**	**0.08**
	Overall		**0.73**	**0.13**	**0.83**	**0.09**	**0.89**	**0.06**

	Document category	Document content	$\mu(PT)$	$\sigma(PT)$	$\mu(E)$	$\sigma(E)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(CA)$	$\sigma(CA)$	$\mu(F)$	$\sigma(F)$
MD	Manuscript	One font and graphics	0.97	0.02	0.19	0.07	0.97	0.01	0.95	0.03	0.97	0.02	0.96	0.02
		Two fonts and graphics	0.95	0.06	0.19	0.24	0.65	0.17	0.61	0.22	0.73	0.24	0.62	0.19
		Only two fonts	0.99	0.02	0.05	0.08	0.50	0.01	0.50	0.07	0.76	0.35	0.50	0.04
		Overall	**0.97**	**0.03**	**0.14**	**0.13**	**0.71**	**0.06**	**0.69**	**0.11**	**0.82**	**0.20**	**0.69**	**0.08**
	Printed	One font and graphics	0.82	0.18	0.47	0.41	0.68	0.19	0.60	0.16	0.82	0.18	0.62	0.13
		Two fonts and graphics	0.85	0.10	0.37	0.23	0.62	0.12	0.64	0.15	0.62	0.12	0.67	0.23
		Only two fonts	0.77	0.11	0.71	0.20	0.67	0.16	0.57	0.10	0.71	0.14	0.60	0.10
		Overall	**0.81**	**0.13**	**0.52**	**0.28**	**0.66**	**0.16**	**0.60**	**0.14**	**0.72**	**0.15**	**0.63**	**0.15**
	Overall		**0.89**	**0.08**	**0.33**	**0.21**	**0.69**	**0.11**	**0.65**	**0.13**	**0.77**	**0.18**	**0.66**	**0.12**

Chapter 6.

A structural signature based on texture for book page characterization

This chapter presents a structural signature based on texture, used for digitized historical book page characterization. The proposed signature is based on varying low-level features (*i.e.* texture, shape, geometric and topological descriptors) and a structural signature. It provides a topological signature of digitized historical book page according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the historical document image content.

Contents

6.1. Introduction

The work conducted in this chapter proposes an automatic characterization approach of DHB pages. The characterization of a DHB page content is based on topological description involving texture, shape and geometric features of elements of content. This characterization is embedded in what we call a structural signature of a HDI. Generating a structural signature for each analyzed DHB page is carried out in three stages: the first step consists in refining the obtained pixel-labeling results (*cf.* Section 5.3.2.2, block 1, Figure 5.1) by taking into account the topological or spatial relationships between pixels, the second one aims to extract homogeneous regions and the third one is generating a structural signature of the page layout and content.

First, to refine the pixel-labeling results, the topological relationship between the selected foreground pixels is introduced by integrating a multi-scale analysis of the topological relationship between pixels.

Secondly, the homogeneous region extraction is performed by combining several points related to texture-based and classical segmentation methods, that have been reported separately in the literature. The extraction of homogeneous regions is based on texture features, multi-scale analysis, an adaptive run-length smoothing algorithm (ARLSA), CC analysis technique and majority voting approach.

Finally, having extracted homogeneous regions, the topological relationships between regions in each page are used to construct a texture-based structural signature in the form of a graph. The obtained signature defines both the spatial organization of the extracted homogeneous texture regions and the different attributes that characterize those regions.

The proposed DHB page signature extraction process is independent of the layout and content of the analyzed DHB pages, and hence, it is applicable to a large variety of HDIs. Indeed, it does not assume *a priori* knowledge regarding page content and structure.

The remainder of this chapter is organized as follows: Section 6.2 reviews the different techniques, used to ensure an automatic characterization approach of DHB pages based on structural signatures, with a particular focus on those related to DIA and HDIA. Section 6.3 presents the proposed approach to generate a structural signature for DHB page characterization. In Section 6.4, we discuss the obtained performance of each step of the proposed structural signature for DHB page characterization by computing several accuracy metrics. Qualitative results are also given to demonstrate the performance of the proposed approach. Our discussion and conclusions are presented in Sections 6.5 and 6.6, respectively.

6.2. Related works

This section reviews the different techniques, used to ensure an automatic characterization approach of DHB pages based on structural representations, with a particular focus on those related to DIA and HDIA. First, within this succinct review, the related works on the post-processing approaches for DI segmentation refinement are discussed in Section 6.2.1. Then, various approaches for region extraction are described briefly in Section 6.2.2. Finally, a short review of the proposed topological representation formalisms in pattern recognition fields is presented in Section 6.2.3.

6.2.1. Post-processing approaches for segmentation refinement

For the segmentation result refinement, many researchers have introduced the topological or spatial relationships between pixels which have not been considered when the extracted texture features have been analyzed (*i.e.* the extracted texture features used in the pixel-clustering task). The introduction of the topological or spatial relationships between pixels as a supplementary task has the advantage to deal with the non-smoothed boundaries due to the texture feature extraction from small pre-defined windows [406]. Based on Chang and Kuo [406], the use of a post-processing labeling task is justified by the two following reasons. Firstly, by using a post-processing labeling

step (e.g. median filtering and morphology-based approaches), the topological or spatial relationships between pixels are integrated which have not been considered when texture features have been analyzed. Secondly, Chang and Kuo [406] confirmed that the texture feature extraction task from small pre-defined windows is not a relevant choice since this technique can generate non-smoothed boundaries. Thus, by introducing the topological or spatial relationships between pixels, the segmentation quality or performance would be improved since the variation produced by the pre-defined sizes of analysis windows can be reduced.

Several methods have been proposed for the segmentation result refinement. For instance, Kumar *et al.* [217] used the MRF technique as a post-processing task to exploit contextual information for the refinement of text extraction and DI segmentation after analyzing the wavelet features. Etemad *et al.* [407] performed a weighted majority voting technique in the decision integration scheme in order to identify text, images and graphical areas from DIs after investigating the wavelet packets technique. Chang and Kuo [406] applied the median filters to smooth the wavelet-based segmentation results. For DI segmentation, Jain *et al.* [188] introduced the CC analysis technique to obtain smoothed text rectangular blocks after analyzing the Gabor features. The morphological dilation operators were applied to connected isolated text edges which were extracted by the discrete wavelet transform for text localization [297]. For a multi-scale segmentation of unstructured DI pages, Etemad *et al.* [249] applied the morphological operations (closing operations) on the image regions after analyzing the wavelet-based features in order to eliminate noise or outliers from the segmented regions. Palfray *et al.* [408] integrated the majority voting technique in order to refine the MRF-based segmentation of ancient newspapers. Charrada and Ben Amara [238] combined several morphological operations (e.g. erosion) and the CC analysis technique as a post-processing phase after using GFs for the extraction of different kinds of nets, such as slightly erased lines or lines with inclinations and curvatures from printed ancient periodicals.

6.2.2. Classical approaches for region extraction

The proposed approaches for homogeneous region extraction have been highly varied in pattern recognition fields. They depend on the type and complexity of the analyzed patterns. For instance, Rais *et al.* [409] categorized the approaches used for textual region extraction into four different categories. The first class of approaches used for text detection and localization is based on using stroke information which ensure the analysis of the intrinsic properties of text. The second class uses an edge-based approach to detect and localize textual regions, by analyzing strength, density or distribution information from edges. The edge-based methods are fast and have good performance if they have high contrast differences between the text and background. The third class is based on CC analysis technique. The use of CC analysis technique ensures the identification of homogeneous regions after filtering the non-text CCs based on defined geometrical constraints. Rais *et al.* [409] stated that the CC analysis methods are robust to font size, but are sensitive to noise. The last class of approaches used for text detection and localization is based on investigating texture features. By assuming that text regions are characterized by specific texture patterns, the analysis of texture descriptors able to segregate textual regions from the background. Rais *et al.* [409] asserted that the texture-based methods are insensitive to noise and low-quality images, but they are time consuming. Gatos *et al.* [410] categorized the algorithms for page segmentation and region extraction into three classes: those based on the smearing and labeling regions, the image profiling in various directions and texture information. They stated that all proposed techniques in the literature can not achieve interesting results for automatic page segmentation and analysis of newspaper pages due the particularities of newspaper pages (*i.e.* haphazard layout of newspaper articles and their close contact). Thus, they proposed to adapt the classical methods of page segmentation and to combine them by presenting a complete system based on gradual extraction of page components for automatic newspaper archive page analysis.

In the literature, the issue of analyzing and segmenting HDIs has been tackled by using the classical approaches based on strong *a priori* knowledge (e.g. RLSA, CC analysis, projection

profiles, morphology). These classical methods presented in the literature which are mainly designed for particular contemporary DIs, address various issues and have many limitations in the case of complex and varied DI corpus and HDIs (*cf.* Section 3.3.1.2). Thus, researchers specialized in DIA have recently addressed many challenges based on few innovative aspects to adapt the proposed classical approaches based on strong *a priori* knowledge for historical DIA. The most popular among these classical approaches for the case of DIs and particularly HDIs, are those based on data-driven or bottom-up strategies of analysis, such as morphology-based, CC analysis and RLSA techniques. These data-driven or bottom-up techniques have been used for DI segmentation for the goal of identifying homogeneous or similar content regions. For instance, Kim and Kim [175] segmented DIs and classified the extracted regions (text, picture, table and graph) using principle component analysis (PCA) algorithm based on analyzing texture features extracted from the GLCM and using the closing operation. Usually, these techniques use heuristic thresholds or rules to determine values of the smoothing values, spatial relationships of the extracted CCs or the structuring element size.

However, many methods have been proposed to estimate automatically these thresholds or rules used for extracting homogeneous or similar content regions in DIs. For instance, Papamarkos *et al.* [411] proposed an unsupervised technique to estimate the proper values of the smoothing variables by calculating the distributions of the black and white run-lengths. The proposed method is based on determining the global maximums of the histogram of horizontal and vertical black run-lengths to estimate the mean character length and height, respectively. Sun [412] proposed a modified version of the RLSA, known as selective CRLA, for Manhattan and non-Manhattan layout DI segmentation. The proposed selective CRLA was performed on a labeled DI after using a CC labeling algorithm [413]. The labeled DI which is derived from the input DI, was used to assign certain labels to the foreground CCs according to their size. Three labels were defined according to the heights of the foreground CCs for text region extraction.

Konidaris *et al.* [414] used run-length smoothing in the horizontal and vertical directions for word segmentation in historical machine-printed DIs. By using dynamic parameters which depend on the average character height (*i.e.* the horizontal run-length threshold is experimentally defined as 50% of the average character height, while the vertical run-length threshold is experimentally defined as 10% of the average character height), the RLSA is adapted to segmenting historical machine-printed DIs into words. The average character height was estimated by computing the maximum value of the histogram with the heights of the CCs of random selected pixels. To group homogeneous textual regions in historical and degraded machine-printed DIs, Nikolaou *et al.* [127] proposed to combine the CC analysis technique with an adaptive RLSA to overcome the drawbacks of the original RLSA (e.g. grouping inhomogeneous regions or different slanted text lines) [101, 102]. The proposed ARLSA is adapted to DIs containing characters with variable font size (*i.e.* it can segment large and small characters). The several thresholds and rules used in their approach was defined according to the geometric properties of neighboring CCs.

Rais *et al.* [409] proposed an accurate text detection and localization method in images based on stroke information and the adaptive RLSA proposed by Nikolaou *et al.* [127]. Ferilli *et al.* [415] used a variant of RLSA which is called run-length smoothing with OR (RLSO) for non-Manhattan document segmentation [416]. The RLSO was processed by using the OR logical operator instead of the AND, to identify irregular CCs. The OR logical operator was applied between the horizontal and vertical smoothing resulting DI, carried out row-wise and column-wise with specified thresholds, respectively. The horizontal and vertical thresholds were defined based on the spacing distribution in the analyzed DI by determining the peaks from the cumulative histogram of run-lengths (*i.e.* the most prominent peak corresponds to the most frequent spacings in the analyzed DI or homogeneous spacings deduced from different DI components). Ferilli *et al.* [415] concluded that the proposed approach gives a satisfactory results when documents are not complex and they are characterized by a uniform text font size. Arora *et al.* [417] proposed a method for Manhattan DI segmentation using dynamic thresholds and identification of each region type by combining the RLSA and recursive top-down technique (*i.e.* projection profiles). The proposed method recursively divided the DI

into hierarchy of homogeneous regions based on investigating the projection profiles after using binarization task, noise removal step and the RLSA to first isolate the headings of the analyzed DI. The threshold values are automatically defined according to the geometric layout of the analyzed DI (*i.e.* physical structure and geometric location of DI regions) by computing distances between horizontal histograms of DI regions. The identification of region types was processed on the basis of several rules deduced from horizontal and vertical histogram values.

Gatos *et al.* [410] combined the RLSA and CC analysis technique to distinguish text and image/drawing regions in old newspaper articles. First, they used the CC analysis technique to assign to every foreground pixel a value according to the height of the box of its connected area for text-tile block separation. Then, the DI was converted to gray-scale and every pixel was classified to either normal text or title according to its gray-scale value. Finally, they proposed a method for text block extraction that is based on the RLSA with adaptive parameters. Gatos *et al.* [133] proposed a segmentation method of historical handwritten DIs into text zones and text lines. For text zone detection, vertical rule lines were detected based on using a fuzzy RLSA [134]. The fuzzy RLSA was used to partition complex handwritten and historical handwritten DIs into textual regions in terms of text words or text lines on the one hand, and graphical regions on the other hand. A fuzzy run-length measure was proceeded for every pixel of the analyzed DI by tracing a background run starting from a background pixel along two directions, to its left and right (this is for horizontal runs, otherwise the up and down directions for vertical runs). On the other hand, vertical white runs and the extracted CCs were afterwards investigated for text line segmentation. LeBourgeois *et al.* [9] proposed a data-driven layout segmentation approach based on the extracted CCs. To localize the main body of the text from Arabic manuscripts, they also estimated the average size of text symbols by computing the average size of all CCs. Then, they computed a text probability value for each extracted CC. Finally, they estimated an automatic threshold for each profile (horizontal and vertical) obtained from the entire image to detect the main body of a text. Ramel *et al.* [72] evaluated various traditional methods used for segmentation of historical printed DIs. They highlighted the limits of the traditional methods to segment HDIs. Thus, they proposed a hybrid segmentation algorithm based on CCs for user-driven page layout analysis of historical printed books.

6.2.3. Topological representation formalisms in pattern recognition fields

To provide an additional structured semantic to the extracted low-level features, the relationships between different objects in images are mainly analyzed. Since a data-driven or bottom-up strategy of analysis has been adopted in this work which is based on low-level data mining of pixels (e.g. texture, position, shape, geometry), we will be focusing on spatial or topological representation formalisms in pattern recognition fields. The characterization of the spatial relationships between objects in images provides a structured semantic that strengthens the low-level techniques of visual content image representation. Brunet *et al.* [418] proposed to describe the spatial relationships between objects contained in images for bridging the gap between low-level or pixel-based descriptors and semantic information. They stated that describing the topological or spatial relationships ensures the integration of strength semantic and enrichment of low-level image processing techniques.

Characterizing and categorizing DIs in the context of DIA is firstly faced by the following challenging fundamental notion of pattern recognition fields: the notion of similarity and distance between two objects or patterns. The similarity measurement is mainly based on the definition and selection of descriptors characterizing the patterns under consideration in the similarity estimation. These descriptors consist of measures, attributes or primitives extracted from the analyzed objects or patterns. They should be relevant to characterize the analyzed objects or patterns by constituting and structuring the feature space in which the similarity estimation will be performed. Then, by measuring mathematically the distance differences between the descriptors of the two analyzed objects or patterns, their degree of similarity can be deduced. Indeed, a pattern “A” is similar

to a pattern "B" if the distances between their descriptors are "small". In fact, in the context of CBIR systems, the idea consists in providing an image query to a developed CBIR system that will retrieves within a database all images similar to the defined one in the query according to a pre-set criterion, based on computing the differences between the image low-level or pixel-based descriptors.

Nevertheless, it is worth noting that the choice of low-level or pixel-based descriptors and distance types has a significant impact when computing the differences between the pattern descriptors to deduce their degree of similarity. Moreover, finding adequate representation formalisms which are able to model the main characteristics of the pattern under consideration, is a crucial task in pattern recognition. These representation formalisms can be of different natures: spatial, temporal or conceptual, *etc.* Hudelot *et al.* [419] stated that image interpretation is a complex task. The existing state-of-the-art approaches for image interpretation are mainly based on strong *a priori* knowledge, and they are dependent on the image type, complexity, *etc.* The image interpretation approaches proposed in the literature has been often criticized for not being generic and the high requirement for adequate *a priori* knowledge acquisition and representation. However, it has been established that the spatial relations between object structures in images play a crucial role for image interpretation and structure recognition. Moreover, they are less prone to variability and complexity of objects in images than the intrinsic characteristics of objects. In addition, they are able to handle with similar appearance objects in images. Nevertheless, when a fine description of spatial relationships is required, it should be probably more appropriate to use the geometric approaches that have the advantage of the image transformation invariance.

Brunet *et al.* [418] categorized the topological or spatial representation formalisms into two classes: implicit and explicit approaches.

- ***Implicit approaches***:
  The implicit approaches produce an overall representation of the existing spatial relationships between all objects in the analyzed image. They are well suited to specific applications, where the goal of image interpretation consists in looking for a spatial configuration of particular objects is required, such as in the fields of face recognition (after face detection task) or medical imaging. Examples of implicit approaches include strings (or 2-D-string such as 2-D C-string [420], 2-D-S-tree [421], *etc.*), trees [422], graphs (e.g. adjacency or neighborhood-based inference strategy [423]), *etc.* For this kind of topological representation formalisms, an inference with respect to the information contained in the representation model is required to deduct all the spatial relationships between objects. In addition, none of these implicit approaches proposes a solution to avoid the complete reconstruction when adding or removing an object in images.

- ***Explicit approaches***:
  On the other side, the explicit approaches produce a detailed structural representation where all the spatial relationships between objects are characterized. They are well suited to more dynamic scenarios where the parts of images or even simply interest objects can be defined by the user. In addition, they are more appropriate for CBIR tools in collections of different types of images, where the use scenarios vary and content may be enriched (e.g. Web images, personal photo albums). Examples of explicit approaches include matrices (e.g. unique bit pattern (UBP) matrix [424]), n-uplet lists (e.g. 9-direction spanning area (9D-SPA) [425]), signature files [426], bin-trees [427], *etc.* For this kind of topological representation formalisms, there are no inferences pertaining to deduct the spatial relationships between all objects in images.

Nevertheless, Brunet *et al.* [418] stated that the most explicit approaches proposed in the literature describing the spatial relationships between pairs of objects in images can be represented by an attributed relational graph (ARG) [428]. Thus, by using the ARG formalism, the similarity

between two images can be computed and subsequently the graph or sub-graph-matching can also be approached.

In our view at an operational level, the topological or spatial representation formalisms integrate a range of topics related to knowledge management process from high-level using ontology towards low-level using image segmentation. In fact, they can be classified into several categories: ontology, statistical and structural representations.

6.2.3.1. Spatial ontology representation

Spatial ontology is considered as a conceptual representation used to model spatial or topological relationships between objects in images and subsequently to produce a knowledge-base of extracted regions. Clementini and Laurini [429] stated that using spatial ontology is required to deal with multiple geometric representations and contextual information. The use of spatial ontology contributes to reducing the semantic gap between the low-level data and thesaurus semantics.

For instance, De La Heras [430] proposed a knowledge-based model for visual understanding of architectural drawing documents. The proposed model is based on an ontological definition of the domain and real data to perform contextual reasoning and detect semantic inconsistencies within the data. Three main tasks are performed to construct this model: First, symbols from the architectural drawing documents are detected. Then, the structural relations between these symbols are extracted. Finally, the modeling of the knowledge that permits the extraction of the semantics is performed using an ontological definition of the domain and real data. De La Heras [430] stated that the ontological definition was nor a straightforward step neither a fast task due to the multiple inconsistencies found in the architectural drawing document database.

Coustaty *et al.* [431] introduced an ontology-based approach on images of drop-caps to deduce automatically semantic information from pixel data. A drop cap is described by a set of extracted regions. The regions were extracted by segmenting a drop cap into main shapes using automatic image processing algorithms proposed in [432, 15]. In fact, the proposed ontology is developed on the basis of a number of low-level features (e.g. area, eccentricity, color) computed from extracted image regions of drop caps and their spatial relationships (e.g. spatial position such as the center of the image or near the sides). In addition, it combined the knowledge of historians using semantic information describing the drop cap content. The goal of the proposed ontology consists in deducing an automatic annotation of semantics of a region using low-level features.

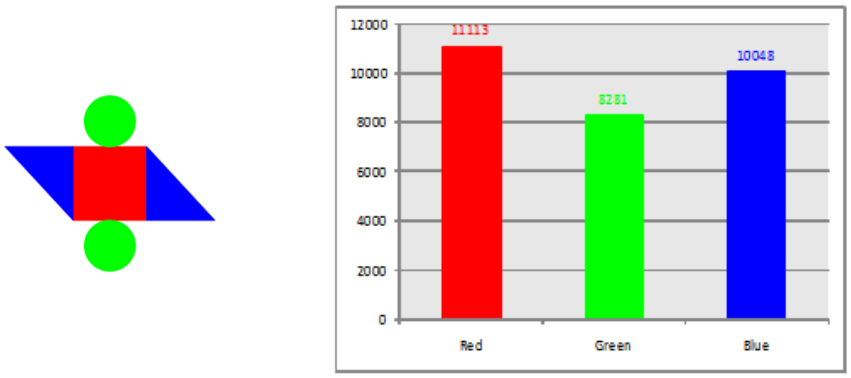
Hudelot *et al.* [433, 419] proposed an ontology of spatial relations enriched by fuzzy representations of concepts, in order to guide image interpretation and recognize the structures that it contains. The recognition of the structures in images is based on structural information on the spatial arrangement of these structures. The fuzzy representations of concepts ensure the definition of the structure semantics (*i.e.* the semantics of the spatial relations) and characterize the relation between these concepts (which are often expressed in linguistic terms) and the low-level information that can be extracted from images. The goals of the fuzzy spatial relation ontology consist in deducing spatial reasoning operations in the images and guiding image interpretation tasks (e.g. localization of objects, segmentation, recognition).

However, the importance of semantics in images has been highlighted in different domains and a growing interest in spatial ontology representation can be observed due the generecity of all concepts that are included in an ontology, modeling complex spatial relations with ontology is still an open question. In addition, semantics can vary a lot depending on the context. Moreover, it is quite difficult to define the proper semantics and the associated fuzzy representation [433, 419].

6.2.3.2. Statistical and structural representations

Two major families of approaches related to feature representation have been reported in pattern recognition literature, statistical and structural representation approaches. Particularly for DI representation, the statistical and structural approaches are broadly applied [14, 15].

1. ***Statistical representations***
   In a statistical representation, each pattern is represented by a N^f-D feature vector. Indeed, a feature vector (V^f) is considered as a point in a real ($\Re$) feature vector space of N^f dimensions (*i.e.* $V^f = (x_1, \ldots, x_{N^f}) \in \Re^{N^f}$). Then, a pattern is characterized by N^f-D feature vector. For example, if our goal is extracting color characteristics of an image, a multi-dimensional histogram of the distribution of color in an image which is called color histogram. The color histogram is performed to characterize a colorimetric space of representation of the colors in three dimensions called space RGB (red, green and blue color space) [434]. A color histogram of an image is a vector with N^f elements $(x_1, \ldots, x_i, \ldots, x_{N^f})$, where x_i denotes the number of pixels having the color i. As shown in Figure 6.1(b), the color histogram of the image in Figure 6.1(a) is represented with three colors (red, green and blue). Indeed, the feature vector $V^f = (11113, 8281, 10048)$ characterizes the image in Figure 6.1(a), based on color descriptors.



(a) Original image (b) Color histogram

Figure 6.1.: Example of a statistical representation of a pattern using a feature vector based on color descriptors ($V = (11113, 8281, 10048)$).

Then, a pattern classification task can be processed using a clustering technique on the computed feature vectors. The idea consists in dividing the N^f-D space into disjoint regions in such a way that each region represent a different class pattern. Jain *et al.* [149] summarized and compared some of the well-known statistical approaches used in various stages of a pattern recognition system. We have already used in Chapters 4 and 5 statistical representations to group pixels sharing similar texture characteristics.

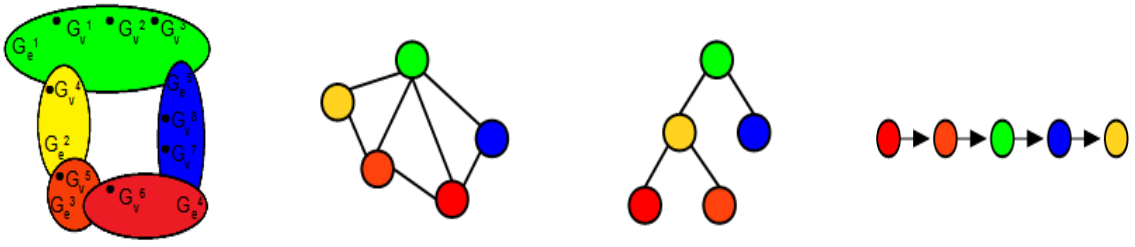
It is worth noting that the use of feature vectors as statistical representations has several significant advantages. The different forms of distances between vectors and a set of mathematical tools or notions available in a vector space (e.g. computing the sum, product, mean, median, center) can be exploited and investigated in a feature vector space. In addition, a large number of clustering techniques, neural networks and decision theoretic methods with low computational complexity of algorithms that can use feature vectors in the case of statistical pattern recognition. Nevertheless, the use of feature vectors as statistical representations has numerous drawbacks. Mainly, in pattern recognition application, similar pre-defined sizes of vectors is usually needed, regardless the nature, size and complexity of the analyzed patterns. Moreover, with using feature vectors as statistical representations, the spatial, topological or binary relationships between the different components of the analyzed patterns that might exist, can not be characterized or described. Hence, separate values of features are only considered in a statistical approach [14].

2. ***Structural representations***
   According to the Canadian psychologist Donald Olding Hebb, the human cognitive system

perceives better the "distributed" nature of a representation than a holistic one of a pattern, due the distributed nature of neural representations [435]. The idea consists in using a specific spatial representation which is called "distributed" representation, that many cells of the nervous system can participate for learning and recognition. Extrapolating on this idea, the use of a structural approach based on decomposing a pattern into a number of distinct entities or simple separate components, is important for pattern recognition fields.

In a structural approach, each pattern is represented by a structure. The objective of a structural approach consists in defining a generic structural representation ensuring the characterization of the distinct simple entities composing a pattern and the spatial or topological relationships between them. Among the most important and best known of structural representations in pattern recognition fields, we mention as examples hyper-graphs, graphs, trees and strings. The various kinds of data structures referred to as trees and strings are algorithmically considered as special cases of graphs [436, 437, 438, 439]. Hyper-graphs, graphs, trees and strings are considered as useful symbolic data structures or structural representations used in pattern recognition fields (*cf.* Figure 6.2):

- An ***hyper-graph*** is considered as the most general formalism of structural representations in pattern recognition. It is a generalization of a graph in which an edge can connect two or more vertices (*cf.* Figure 6.2(a)).
- A ***graph*** (G) is a well-known formalism of a structural representation in pattern recognition. It describes a complex pattern through the different elementary entities composing it (*i.e.* graph vertices or nodes) and the relational properties between them (*i.e.* graph edges). Hence, it is composed of a finite set of vertices or nodes, connected by a set of edges (*cf.* Figure 6.2(b)). Vertices or nodes (G_v) represent distinct simple entities composing a complex pattern under consideration. Edges (G_e) represent the relationships between each two entities or parts of the analyzed pattern, where each edge connects two nodes in the graph G (*i.e.* $G_e = (G_v^s, G_v^d)$, such that both G_v^s and G_v^d are two vertices that belong to the set G_v).
- A ***tree*** is a graph in which any two vertices are connected by exactly one path (*cf.* Figure 6.2(c)).
- A ***string*** is a tree whose vertices are connected to at most two other vertices, *i.e.* sequence of vertices (*cf.* Figure 6.2(d)).



(a) Hyper-graph (b) Graph (c) Tree (d) String

Figure 6.2.: Kinds of structural representations.

A structural approach has more developed representational capabilities than a statistical one, since a feature vector can be modeled using a structural representation but not *vice versa* (*i.e.* a vector can be represented by a graph in which nodes or vertices correspond to feature vector elements). The above mentioned limitations of using statistical representations with feature vectors, mainly the size constraint and lacking ability of integrating potential

relationships between distinct simple entities composing a pattern, can be overcome by using graph-based representations. Note that these potential relationships between distinct simple entities composing a pattern can be of different natures (*i.e.* spatial, temporal or conceptual). Indeed, symbolic data can be integrated by means of edges in a structural representation to model spatial or topological relationships between distinct simple entities composing a complex pattern. In addition, the numbers of vertices and edges are neither limited nor predefined, and it can be adapted to the size and the complexity of each individual pattern under consideration. Moreover, two structural representations or graphs with different numbers of vertices and edges can be compared. Conversely, when using statistical representations with feature vectors in comparing two patterns, feature vectors defined in the same dimensional vector space are usually required [14].

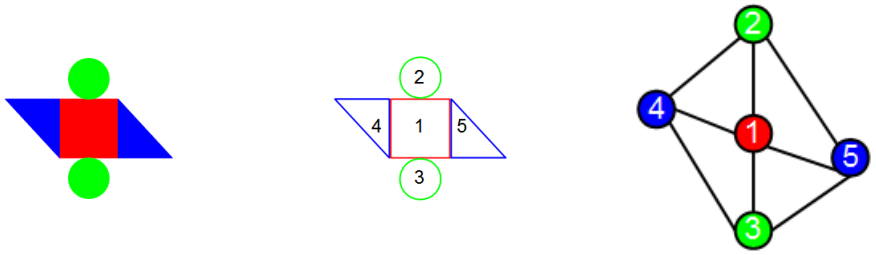
However, the significant increase of the complexity of many algorithms using graph-based representations is considered one serious limitation of using graphs compared to statistical representations using feature vectors. For instance, the comparison of two feature vectors takes linear time with respect to the length of the two vectors, while the comparison of two graphs for isomorphism takes exponential time (*cf.* Section B.8) [440, 441, 439]. But, thanks to the improvement of computer capacities, structural representations have become more and more popular, and they have been intensively used in different fields of pattern recognition and machine vision [441]. For instance, graphs have been studied with emerging interest in the fields of bio-informatics and chemo-informatics [438, 442, 443, 444, 445]. Graph-based representations have proven to be flexible in a wide range of image types [83]. Hence, there is a growing interest in using graph-based representations in many applications of image recognition and classification, thanks to the inherent flexibility, generality and ability of graphs to represent both properties of entities and their potential spatial or topological relationships [446, 447, 448]. Recently, several graph-based applications have been developed on different fields, such as chemistry, Web and image-related tasks like image classification and retrieval [449]. A number of works based on structural approach have been proposed for fingerprint classification [450, 451] and diatom identification [452, 453]. Another field of research where graphs have been recently investigated and examined to detect network anomalies and predict abnormal events [454, 455, 456].

The use of the structural representations or graphs is not new for the DIA community. Numerous studies based on graphs have been proposed for different kinds of DIs (e.g. HDIs, contemporary DIs, graphical DIs such as maps, flowcharts, electrical, architectural and engineering drawings). Some studies looked at the whole DI, while others examined predefined part of them, such as the graphical images (e.g. drop caps [15, 83], symbols [81]) or the textual parts (e.g. characters [457], words [82]). These research studies have been proposed for various DIA tasks (e.g. segmentation, indexing, spotting, retrieval, recognition, analysis). For instance, graph-based structures have been used for graphical symbol [458, 459] and character [460, 461] recognition. Moreover, graphs have found widespread applications in Web documents [462, 463, 464].

In the context of the NaviDoMass project, Jouili *et al.* [15] proposed a structural-based framework for drop cap clustering based on a graph-matching task. They proposed a graph-based representation for drop caps, and they compared their proposed representation with a statistical one based on the generic Fourier descriptor (GFD). They concluded that the results provided from the use of the GFD are the lower ones, and the structural-based representation is more appropriate to handle images of drop caps than the statistical one.

In literature, many research studies investigated graph-based structures to represent an image with a graph [465, 466, 467]. For instance, in Figure 6.3(c), an image (*cf.* Figure 6.3(a)) is represented by a graph based on region-based segmentation approach (*cf.* Figure 6.3(b)). In a graph-based representation, vertices (G_v) and their attributes (A^v), describe the segmented

regions, while edges (G_e) and their attributes (A^e) describes the interrelationships between the G_v. When additional information is integrated in G_v and/or G_e, the designed graph is considered as attributed. This information is called attributes (A^v and/or A^e). The attributes can be numeric and/or symbolic labels (*i.e.* scalar values characterizing the segmented regions and their interrelationships) or more complex descriptions such as strings or feature vectors. A brief review of the basic definitions and concepts of graphs is presented in Appendix B and particularly in Section B.8.



(a) Original image (b) Segmented image (c) Graph

Figure 6.3.: Example of a structural representation of a pattern using a graph.

6.3. Proposed structural signature for digitized historical book page characterization

Figure 6.9 illustrates the detailed schematic block diagram of the proposed structural signature for DHB page characterization. First, to refine the pixel-labeling results, the topological relationship between the selected foreground pixels is introduced by integrating a spatial multi-scale analysis of majority votes. Secondly, the homogeneous region extraction is processed by combining several points related to texture-based and classical segmentation methods, that have been reported separately in the literature. The extraction of homogeneous regions is based on texture features, multi-scale analysis, an ARLSA, CC analysis technique and majority voting approach. Finally, having extracted homogeneous regions, the topological relationships between regions in each page are used to construct a texture-based structural signature in the form of a graph. The obtained signature defines both the spatial organization of the extracted homogeneous texture regions and the different attributes that characterize those regions.

6.3.1. Pixel-labeling refinement

The pixel-labeling task consists in labeling independently each foreground pixel based on analyzing texture features on different sizes of sliding windows. Nevertheless, due particularly to the presence of noise, the foreground pixels will be prone to incorrect labeling. However, based on the fact that the neighboring pixels have higher probability to belong to the same page content type, the mislabeling errors can be corrected (*i.e.* the neighboring pixels should have the same label, except for the pixels belonging to different page content types.). As a matter of fact, a refinement task of the pixel-labeling results can improve significantly the overall results.

Thus, to refine the pixel-labeling results (*cf.* Figure 6.4(b), Section 5.3.2.2, block 1, Figure 5.1), a first step called *"pixel-labeling refinement"*, is introduced in the proposed algorithm of homogeneous region extraction from HDIs by taking into consideration the topological or spatial relationships between the selected foreground pixels and integrating the spatial multi-scale analysis of majority votes. First, the Euclidean distance (ED) between each computed Gabor feature vector of the selected foreground pixel (V_{p_f}), and the Gabor feature vector of the centroid of the cluster

belonging to it ($V_{p_f}^c$) is calculated (the centroid of the cluster is computed regarding the extracted Gabor features and not regarding the spatial descriptors). Then, the foreground pixels are sorted in descending order according to the computed ED values in such a way that the first processed foreground pixel is the one that has a higher ED value. The higher the value of the computed ED, the higher probability that the foreground pixel is improperly labeled, since it is far from the centroid of the cluster it belongs to. Thus, the first processed foreground pixels are those that have high values of ED by using the multi-scale majority voting technique. By performing the spatial multi-scale approach in the majority voting technique, small isolated groups of mis-labeled pixels will be labeled correctly. For each selected foreground pixel, the pixels defined at each size of sliding windows are categorized and summed according to the label. Then, the maximum value among the cluster labels for the different pre-defined sizes of sliding windows is selected. Indeed, a local decision on the label of each selected foreground pixel is taken using the maximum number or majority of pixel labels which is performed at the same four pre-defined sizes of sliding windows in the texture feature extraction step (*i.e.* (16×16), (32×32), (64×64) and (128×128)). Afterwards, the next processed foreground pixel is one that has a smaller ED value than the former one. The labels of foreground pixels are updated on each run of the multi-scale majority voting technique on each foreground pixel to ensure a relevant refinement of the pixel-labeling results. Since the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs has been performed, a refined pixel-labeled DI is obtained (*cf.* Figure 6.4(c)). Figure 6.4 illustrates the resulting DI derived from the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs.

The *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs is defined according to the algorithm 1. Some steps in the algorithm 1 are shown in red color. This coloring is meant to highlight the main computation steps related to the proposed algorithm for refinement of pixel-labeling results.

Figure 6.5 illustrates the intermediate resulting DIs derived from the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs (*cf.* Figure 6.9) with the spatial multi-scale majority voting technique. We note that the small isolated groups of pixels (blue) have been relabeled as graphical pixels (green).

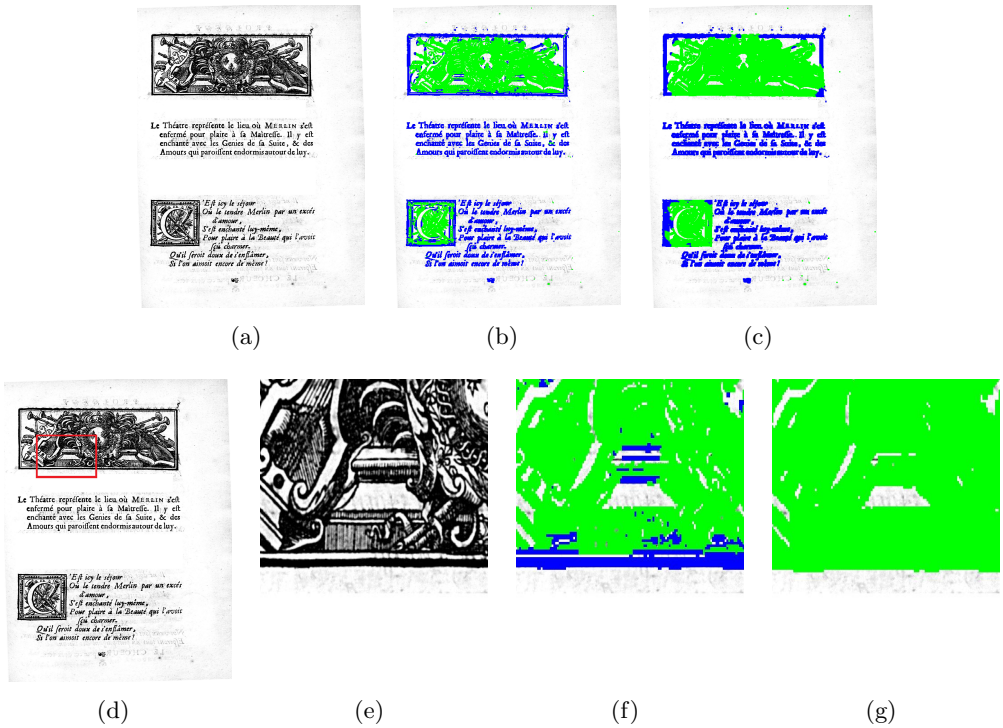
6.3.2. Post-processing

As already seen on the proposed algorithm of homogeneous region extraction from HDIs (*cf.* Figure 6.9), our goal is to find homogeneous regions defined by common characteristics or similar texture features as easily, quickly and automatically as possible. We need to identify few groups of pixels sharing common characteristics or similar textural properties at this stage in order to extract homogeneous regions (*i.e.* to partition text into columns or text blocks, and to identify the graphical regions).

So since the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs has been performed, our output data consists of a refined pixel-labeled DI ($Image_{ref}$, *cf.* Section 6.3.1, Figure 6.6(c)). Thus, the goal of the post-processing step consists in grouping pixels which share common characteristics or similar textural properties from the $Image_{ref}$ to find homogeneous regions from HDIs. The post-processing task is conceptualized by three modular processes:

1. ***Connected component extraction and labeling*** (*cf.* Section 6.3.2.1),
2. ***Color layer separation*** (*cf.* Section 6.3.2.2),
3. ***Adaptive run-length smearing algorithm*** (*cf.* Section 6.3.2.3).

Figures 6.6 and 6.7 illustrate the intermediate results of the different tasks performed to extract homogeneous regions from HDIs.



(a) (b) (c)

(d) (e) (f) (g)

Figure 6.4.: Illustration of the resulting DI derived from the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs, using the auto-correlation features. Figures (a) and (e) show an example of HDI (as an input) and a zoomed region of (a), respectively. Figures (b) and (f) illustrate the pixel-labeled DI of the analysis of the extracted Gabor features (graphical regions (green) and textual regions (blue)) and a zoomed region of (b), respectively (*cf.* Section 5.3.2.2, block 1, Figure 5.1). Figures (c) and (g) depict the outputs of the resulting DI derived from the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs and a zoomed region of (c), respectively. Figure (d) illustrates the selected region (shown with a red color rectangle) of the cropped image (e).

6.3.2.1. Connected component extraction and labeling

We aim in this step to extract and label CCs from the DI under consideration. The labeling of the extracted CCs is performed by retrieving the label of the most represented pixels. A binarization step is firstly performed using a standard parameter-free binarization method, the Otsu's method, on the DI under consideration (*cf.* Figure 6.6(a)) to obtain a binarized DI ($Image_b$, *cf.* Figure 6.6(b)) and subsequently to retrieve the CCs. Then, the majority voting technique is applied on each extracted CC from the $Image_b$ by computing the maximum number or majority of pixel labels belonging to the localized CC on the $Image_{ref}$ (*cf.* Section 6.3.1, Figure 6.6(c)). Using the majority voting technique ensures the labeling of the extracted CCs (CC_b) from the $Image_b$. Therefore, the extracted CC_b from the $Image_b$ are labeled according to the obtained refined pixel-labeling results in the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs ($Image_{ref}$, *cf.* Section 6.3.1, Figure 6.6(c)). The resulting DI derived from the labeling task of the CC_b according to the obtained refined pixel-labeling results in the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs ($Image_{ref}$, *cf.* Section 6.3.1, Figure 6.6(c)) by using the majority voting technique is illustrated in Figure 6.6(d). Since the CC_b are labeled according to the $Image_{ref}$, a pixel-labeled DI is produced ($Image_{mv}$).

Algorithm 1 Refinement of pixel-labeling results

$i \leftarrow 1$
while $i \leq M$ **do**
    Compute the $ED(i)$ between V_{p_f} and $V^c_{p_f}$
    $i \leftarrow i + 1$
Sort the foreground pixels in descending order according to the computed ED
Determine the number maximum of clusters k_{max}
$i \leftarrow 1$
while $i \leq M$ **do**
    $k \leftarrow 1$
    while $k \leq k_{max}$ **do**
        $acc_k \leftarrow 0$
        $j \leftarrow 1$
        while $j \leq N_w$ **do**
            $l \leftarrow 1$
            while $l \leq$ number of pixels in N_w **do**
                if $label(l) = label(k)$ **then**
                    $acc_k \leftarrow acc_k + 1$
                $l \leftarrow l + 1$
            $j \leftarrow j + 1$
        $k \leftarrow k + 1$
    $newLabel \leftarrow label(\max_k(acc_k))$
    if $label(i) \neq newLabel$ **then**
        Update the label: $label(i) \leftarrow newLabel$
    else
        Keep the same label: $label(i)$
    $i \leftarrow i + 1$

where M and N_w denote the number of foreground pixels and number of sliding windows, respectively.

6.3.2.2. Color layer separation

The color layer separation task ensures the classification of the extracted CCs (CC_{mv}) from the $Image_{mv}$ according to their labels (*i.e.* content type). When the CC_{mv} are separated according to their labels, the issues caused by the complex, dense and overlapping document layout of HDIs when grouping pixels will be overcome. The color layer separation task is performed on the $Image_{mv}$ to split the CC_{mv} according to their labels (*i.e.* color). Therefore, a DI containing only single color CCs is generated for each color layer. For instance, in the example illustrated in Figure 6.6, there are two colors representing separately the graphical (blue) and textual (green) CCs in Figures 6.6(e) and 6.6(i), respectively.

6.3.2.3. Adaptive run-length smearing algorithm

The determination of homogeneous regions is based on identifying the largest CCs. As a consequence, by replacing a sequence of background pixels with foreground ones and afterwards grouping pixels which share common characteristics or similar textural properties from the refined pixel-labeled DI, the extraction and identification of homogeneous regions will be more accurate and relevant. Indeed, the idea of this step is to fill automatically the space within each CC to partition text into columns or text blocks on the one hand, and to identify the graphical regions on the other hand.

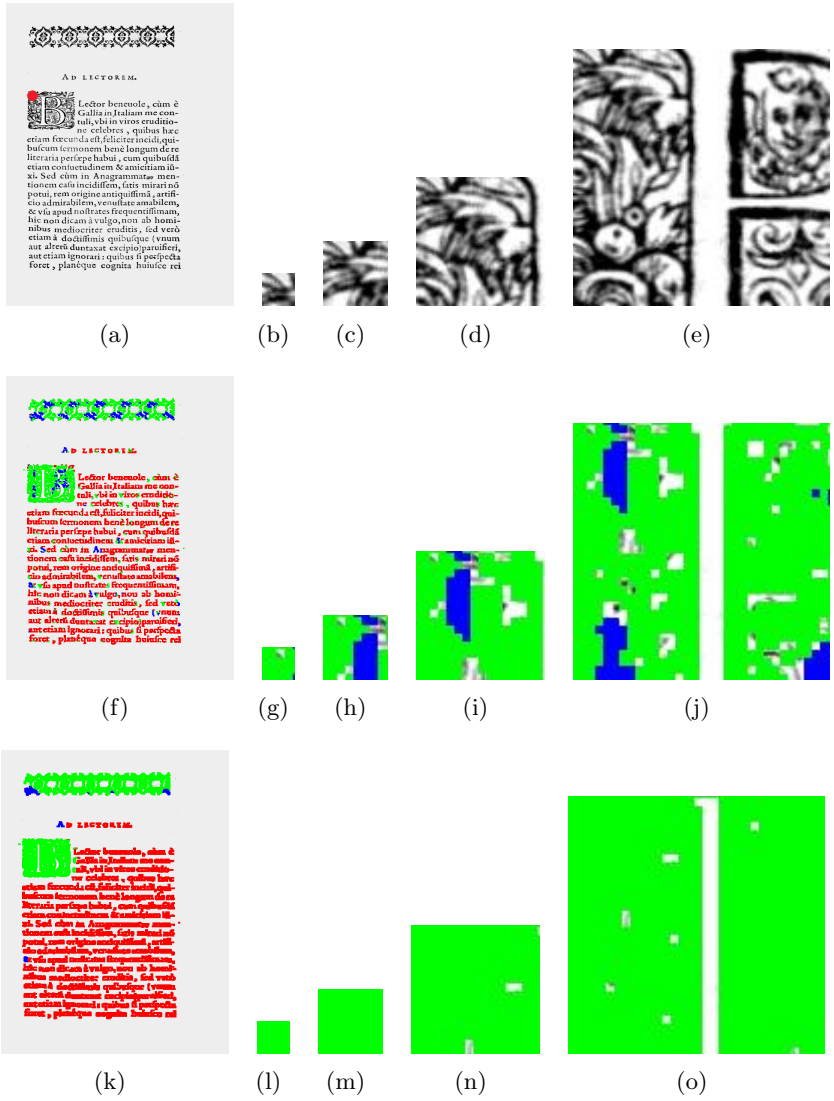


Figure 6.5.: Illustration of the intermediate resulting DIs derived from the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs with the spatial multi-scale majority voting technique and the auto-correlation features. Figures (a), (f) and (k) illustrate an example of HDI (as an input), a pixel-labeled DI and a refined pixel-labeled DI, respectively. The selected spatial position of the cropped images is shown with a red color circle in (a). Figure (f) depicts a refined pixel-labeled DI (the resulting DI derived from the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs). Figures (b), (g) and (l) show (16×16) windows of an input DI, a pixel-labeled DI and a refined pixel-labeled DI, respectively. Figures (c), (h) and (m) show (32×32) windows of an input DI, a pixel-labeled DI and a refined pixel-labeled DI, respectively. Figures (d), (i) and (n) show (64×64) windows of an input DI, a pixel-labeled DI and a refined pixel-labeled DI, respectively. Figures (e), (j) and (o) show (128×128) windows of an input DI, a pixel-labeled DI and a refined pixel-labeled DI, respectively.

So an adaptive RLSA (ARLSA) is proposed in this work which is a modified version of the state-of-the-art RLSA [102]. The RLSA studies the spaces between black pixels in order to link neighboring black areas by applying the run-length smearing both horizontally and vertically by means of a logical *AND* operation. It operates by replacing a horizontal (vertical, respectively) sequence of background pixels with foreground ones if the number of background pixels in the

horizontal (vertical, respectively) sequence is smaller or equal to a pre-defined horizontal threshold (T_h) (vertical threshold (T_v), respectively). In this work, the proposed ARLSA applies the run-length smearing both horizontally and vertically by means of a logical OR operation. The idea of the proposed ARLSA which is based on character size, consists in determining low values of horizontal and vertical thresholds to identify the inter-character and inter-line spaces, respectively. In addition, the proposed ARLSA determines automatically the horizontal (T_h) and vertical (T_v) thresholds which correspond to the run-length smoothing values. The quality of the RLSA results depends on setting the proper thresholds. Setting specific values to the thresholds is a delicate issue since it must be adapted to the peculiar DI layout features. Indeed, too high threshold values can wrongly merge different content blocks of the analyzed DI, while too low ones can produce an over-segmented DI. Therefore, we propose a technique for determining automatically the proper values of T_h and T_v.

To obtain the proper values of T_h and T_v, the two histograms of the widths and heights of the extracted CCs are examined, respectively. These two histograms gives the distributions of the widths and heights of the extracted CCs from a HDI. The estimation of the horizontal threshold (T_h) (vertical threshold (T_v), respectively) is based on the determination of the global maximum of the histogram of the widths of the extracted CCs (GMH_w) (heights of the extracted CCs (GMH_h), respectively). The GMH_w (GMH_h, respectively) gives mainly information about the mean character length (height, respectively). Nevertheless, due particularly to the characteristics of HDIs linked to the presence of noise and degradation caused by copying, scanning and aging (staining, mold or moisture and faded out ink and uneven lighting due to folded and corrugated parchment or papyrus, *etc.*), the two estimated global maximums of the two histograms of the widths and heights of the extracted CCs correspond usually to the width and height of noise CCs. Thus, we need to exclude the CCs corresponding to noise by defining a rule when analyzing these two histograms. Indeed, to link horizontally (vertically, respectively) neighboring black areas, if the global maximum of the histogram of the widths of the extracted CCs (GMH_w) (heights of the extracted CCs (GMH_h), respectively) is smaller to a pre-defined threshold (T_c), GMH_w is equal to T_c (GMH_h is equal to T_c, respectively), otherwise T_h (T_v, respectively) is equal to $c_h \times GMH_w$ ($c_v \times GMH_h$, respectively). Where c_h, c_v and T_c have been experimentally determined, and they are equal to 1.1, 1.5 and 10, respectively. T_c corresponds to the pre-defined threshold which characterizes the CCs corresponding to noise. T_c is used to exclude the CCs corresponding to noise. c_h and c_v are the pre-defined weights for determining the horizontal and vertical thresholds T_h and T_v, respectively. The estimation of the horizontal and vertical run-length smoothing values (T_h and T_v) are defined according to the two algorithms 2 and 3, respectively. Some steps in the two algorithms 2 and 3 are shown in red color. This coloring is meant to highlight the main computation steps related to the proposed algorithms for estimation of horizontal and vertical run-length smoothing values.

Algorithm 2 Estimation of horizontal run-length smoothing value

1: **function** ESTIMATION OF $T_h(GMH_w, T_c)$
2: **if** $GMH_w \geq T_c$ **then**
3: $T_h \leftarrow c_h \times GMH_w$
4: **else**
5: $T_h \leftarrow T_c$
6: **return** T_h

Once the horizontal and vertical run-length smoothing values (T_h and T_v) are estimated automatically according to the DI content (*i.e.* particularly the distributions of the widths and heights of the extracted CCs of the binarized DI), the proposed ARLSA is applied on each binarized resulting DI derived from the color layer separation task. A binarizing step is performed on each resulting DI of the color layer separation task ($Image^l_{mv}$) by using the Otsu's algorithm, to generate a binarized

Algorithm 3 Estimation of vertical run-length smoothing value

```
function ESTIMATION OF T_v(GMH_h, T_c)
    if GMH_h ≥ T_c then
        T_v ← c_v × GMH_h
    else
        T_v ← T_c
    return T_v
```

resulting DI of the color layer separation task ($Image_{b^l_{mv}}$). The proposed ARLSA operates by taking the logical OR of the horizontally ($Image^h_{RLSA}$, *cf.* Figure 6.6(f) (6.6(j), respectively)) and vertically ($Image^v_{RLSA}$, *cf.* Figure 6.6(g) (6.6(k), respectively)) merged images of each resulting DI of the color layer separation task to generate Figure 6.6(h) (6.6(l), respectively). The proposed ARLSA is defined according to the algorithm 4. After applying the proposed ARLSA (*cf.* 4) on each binarized resulting DI of the color layer separation task ($Image_{b^l_{mv}}$), the logical NOT is performed on each resulting DI to merge the different resulting DIs derived from the application of the ARLSA task (*cf.* Figures 6.6(h) and 6.6(l)) with the logical OR. Since the merge process with the logical OR of the different resulting DIs derived from the application of the ARLSA task (*cf.* Figures 6.6(h) and 6.6(l)) has been performed, a binarized post-processed DI is generated ($Image_{b,post}$, *cf.* Figure 6.7(a)) in which the neighboring black areas are linked by applying the run-length smearing both horizontally and vertically. The labeling step of the extracted CCs (CC_{post}) from the resulting DI derived from the application of the merge process with the logical OR ($Image_{b,post}$, *cf.* Figure 6.7(a)), is performed by taking into account the deduced labels from Figure 6.6(d) ($Image_{mv}$). Indeed, the post-processed pixel-labeled DI ($Image_{post}$, *cf.* Figure 6.7(b)) is obtained by labeling the CC_{post} according to the deduced labels from Figure 6.6(d) ($Image_{mv}$) and by using the majority voting technique. Some steps in the algorithm 4 are shown in red color. This coloring is meant to highlight the main computation steps related to the proposed ARLSA.

Algorithm 4 Adaptive run-length smearing algorithm

1: Extract the $CC_{b^l_{mv}}$ from the $Image_{b^l_{mv}}$
2: Estimate the T_h value
3: Estimate the T_v value
4: Generate the $Image^h_{RLSA}$ by performing the **RLSA** on the $Image_{b^l_{mv}}$ in the **horizontal direction** using T_h
5: Generate the $Image^v_{RLSA}$ by performing the **RLSA** on the $Image_{b^l_{mv}}$ in the **vertical direction** using T_v
6: Apply the logical **OR** of the $Image^h_{ARLSA}$ and $Image^v_{ARLSA}$

6.3.3. Homogeneous region extraction

Finally, the homogeneous or similar content regions are extracted and labeled by extracting the CC_{post} from the $Image_{post}$ (*cf.* Figure 6.7(b)) in order to identify few groups of pixels sharing common characteristics or similar textural properties (*i.e.* text is partitioned into columns or paragraphs, and graphical regions are localized) (*cf.* Figure 6.7(c)). To define an extracted region, a bounding box covering all the pixels belonging to the extracted CC is used (*i.e.* a contour tracking of the shape of the extracted CC is carried out to identify the bounding box from each component). Then, the colors of the external contours of the defined bounding box is drawn according to the label deduced from the resulting DI of using the majority voting technique (*cf.* Figure 6.6(d)).

However, as already mentioned, due particularly to the characteristics of HDIs linked to the presence of noise and degradation, many extracted CCs correspond usually to noise (*cf.* Figures 6.8(d) and 6.8(e)). Indeed, if we take into consideration all the extracted CC_{post} from the post-processed

pixel-labeled DI ($Image_{post}$), many non-significant CCs can be extracted and subsequently irrelevant regions with small sizes can be identified (*cf.* Figures 6.8(d) and 6.8(e)). Thus, a selection of only the most representative homogeneous regions or CCs (CC^{rep}_{post}) from the extracted CC_{post} in the post-processed pixel-labeled DI ($Image_{post}$, *cf.* Figure 6.8(c)) is required. This step is necessary to ignore small isolated CCs corresponding to noise regions as possible for the subsequent processing steps. Figure 6.8 illustrates the significant role of the CC selection step to retrieve only the representative homogeneous regions. We note that the small isolated CCs have not been retrieved (*cf.* Figures 6.8(f) and 6.8(g)). The idea of the selection of representative CCs consists in retrieving only significant or representative homogeneous regions by ignoring small isolated CCs which have sizes lower than 5% of the total number of pixels of all extracted CC_{post} ($S_{CC_{post}}$) in the post-processed pixel-labeled DI ($Image_{post}$, *cf.* Figure 6.8(c)). The selection of representative information in terms of number of pixels is based on respecting a constructive compromise between having representative CCs and ignoring many CCs which correspond to noise and that complicate subsequent processing steps. The task of the selection of representative homogeneous regions or CCs (CC^{rep}_{post}) from the extracted CC_{post} in the post-processed pixel-labeled DI ($Image_{post}$, *cf.* Figure 6.8(c)) is defined according to the algorithm 6. This step ensures both the size reduction of the proposed graph-based signatures and the speeding up their handling by graph algorithms whose computational complexity is exponential in the number of vertices of the involved graphs.

Figure 6.10 illustrates the detailed schematic block representation of the proposed algorithm of homogeneous region extraction from HDIs. The proposed algorithm of homogeneous region extraction from HDIs is defined according to the algorithm 5. Some steps in the two algorithms 5 and 6 are shown in red color. This coloring is meant to highlight the main computation steps related to the proposed algorithms for extraction of homogeneous regions from HDIs and selection of representative CCs.

Algorithm 5 Extraction of homogeneous regions from HDIs

1: Extract the CC_b from the binarized $Image_b$
2: Generate the $Image_{mv}$ by performing the majority voting technique (CC_b,$Image_{ref}$)
3: Apply the color layer separation technique on the $Image_{mv}$
4: Determine the number of different labels l_{max}
5: Extract the CC_{mv} from the $Image_{mv}$
6: $l \leftarrow 1$
7: **while** $l \leq l_{max}$ **do**
8: Retrieve the CC^l_{mv} from the $Image_{mv}$
9: Generate the $Image^l_{mv}$ corresponding to the retrieved CC^l_{mv}
10: Generate the $Image_{b^l_{mv}}$ by binarizing the $Image^l_{mv}$
11: Generate the resulting DI $Image_{b^l_{mv}}$ derived from the application of the proposed **ARLSA**
12: Generate the $Image^l$ by applying the logical **NOT** on the $Image_{b^l_{mv}}$
13: $Image_{b,post} \leftarrow Image_{b,post}$ **OR** $Image^l$
14: $l \leftarrow l + 1$
15: Extract the CC_{post} from the $Image_{b,post}$
16: Select the representative CC^{rep}_{post} from the CC_{post}
17: Generate the $Image_{post}$ by performing the majority voting technique (CC^{rep}_{post},$Image_{mv}$)
18: Extract and label homogeneous regions from the $Image_{post}$

where CC^{rep}_{post} denote the selected representative homogeneous regions from CC_{post}.

Algorithm 6 Selection of representative CCs

Sort the extracted CC_{post} by the number of pixels and by descending order
$pacc1 \leftarrow 0$
$i \leftarrow 1$
while $i \leq N_{CCs}$ **do**
    $pacc1 \leftarrow S_{CC^i}$
    $i \leftarrow i + 1$
$T^1_{CCs} \leftarrow 0.95 \times pacc1$
$T^2_{CCs} \leftarrow 0.05 \times pacc1$
Keep the largest extracted CC (CC^1) having as size S_{CC^1}
$pacc2 \leftarrow S_{CC^1}$
$j \leftarrow 2$
while $j \leq N_{CCs}$ & $j \geq 2$ **do**
    if $((S_{CC^j} \geq T^2_{CCs}$ AND $S_{CC^j} + S_{CC^{j+1}} \geq T^1_{CCs})$ OR $(pacc2 \leq T^1_{CCs}))$ **then**
        $pacc2 \leftarrow pacc2 + S_{CC^j}$
        Keep the CC^j
    $j \leftarrow j + 1$

where N_{CCs} and S_{CC^i} denote the number of the extracted CC_{post} and the number of pixels belonging to the CC^i, respectively.

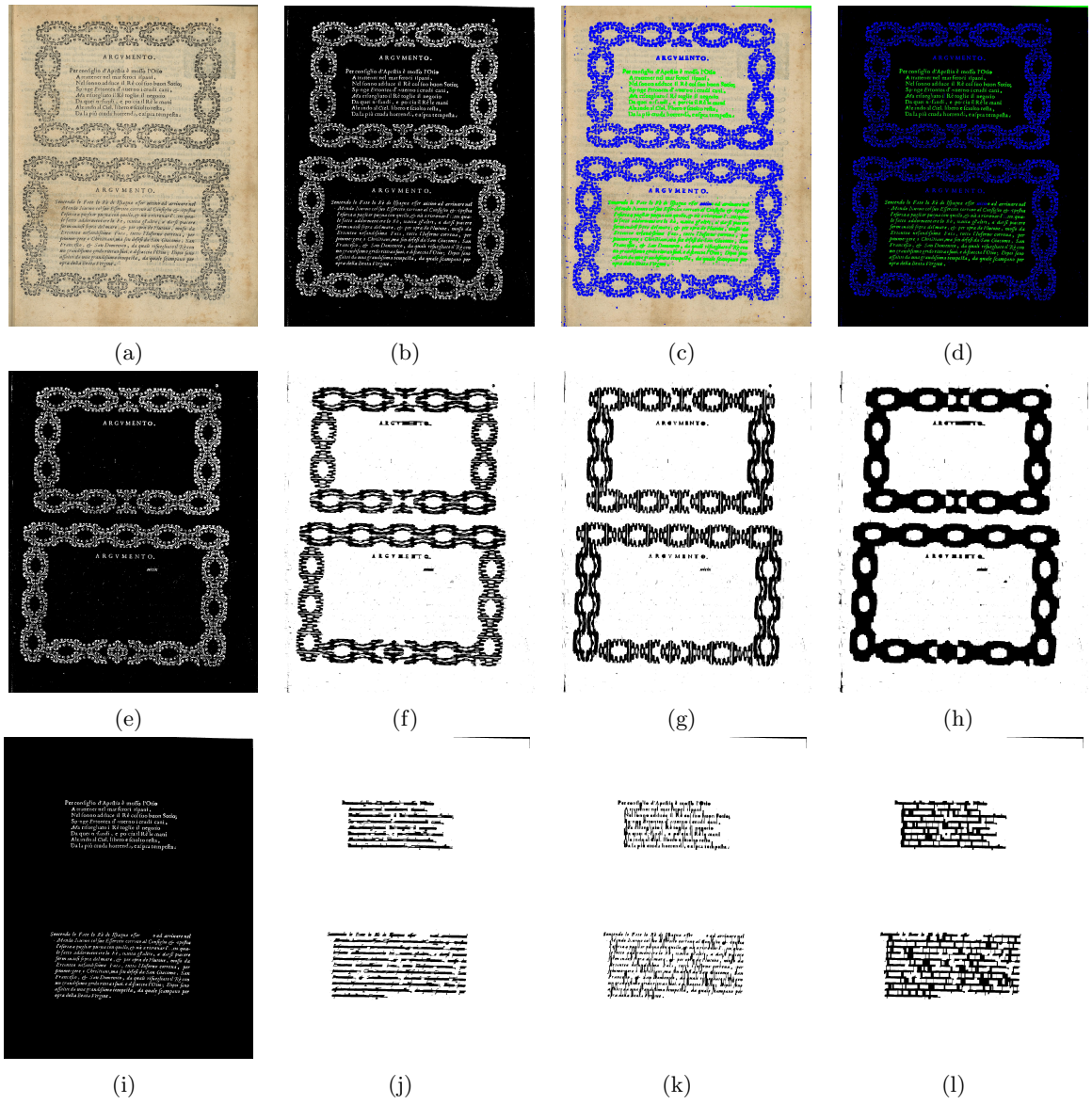
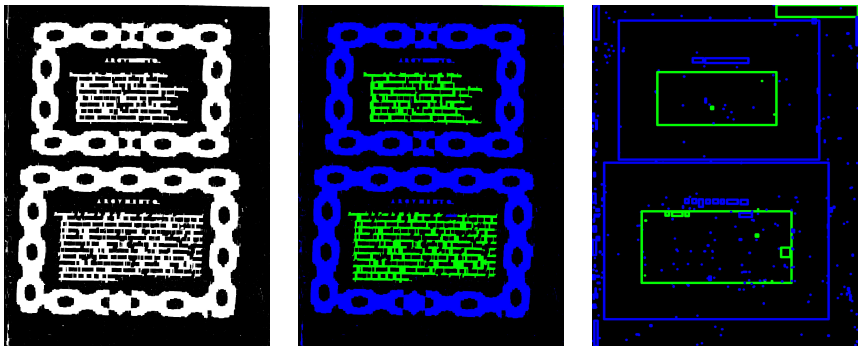
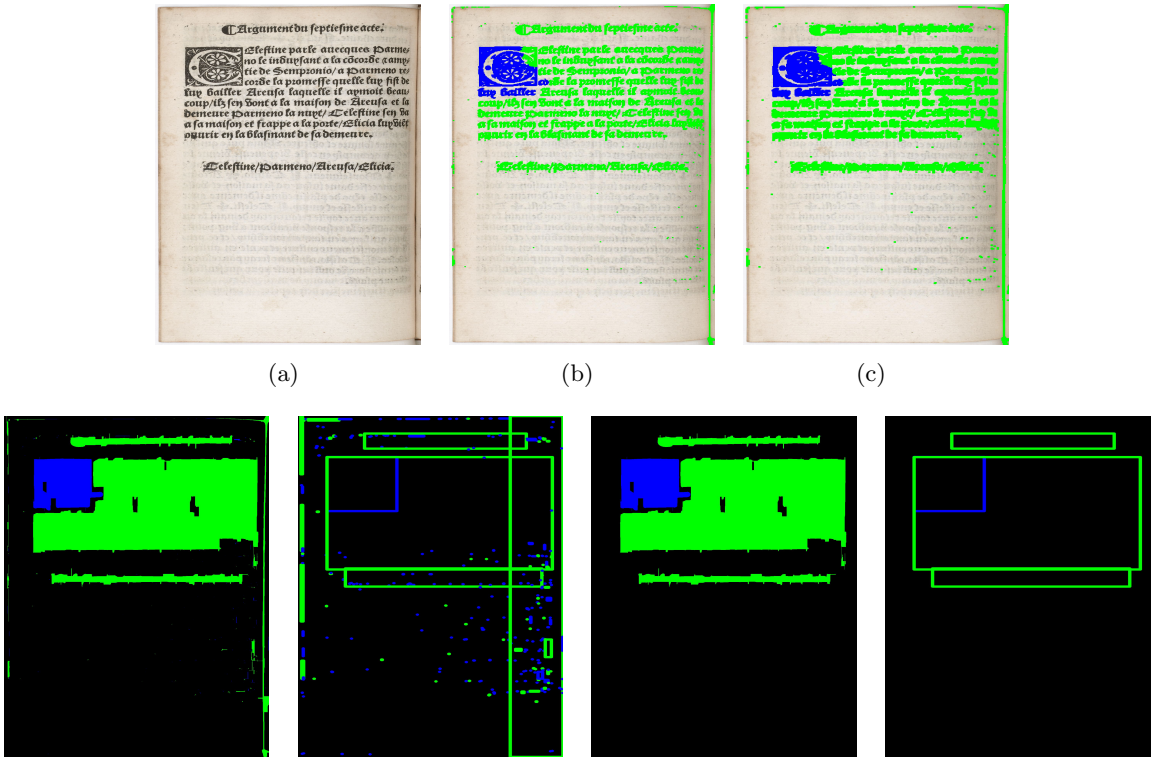


Figure 6.6.: Illustration of the first intermediate results of the different tasks performed for homogeneous region extraction from HDIs, using the Gabor features. Figure (a) illustrates the original HDI. Figure (b) illustrates the binarized HDI of Figure (a). Figure (c) shows the resulting DI derived from the *"pixel-labeling refinement"* step. Figure (d) shows the resulting DI derived from labeling the extracted CCs from the binarized DI according to the obtained refined pixel-labeling results in the *"pixel-labeling refinement"* step by using the majority voting technique. Figures (e) and (i) are the two resulting binarized DIs of the color layer separation task, illustrating separately the graphical (blue) and textual (green) CCs, respectively. Figures (f) and (g) show the resulting DIs of the application of the run-length smearing both horizontally and vertically on the resulting binarized DI representing the graphical regions (*cf.* Figure (e)), respectively. Figures (j) and (k) show the resulting DIs of the application of the run-length smearing both horizontally and vertically on the resulting binarized DI representing the textual regions (*cf.* Figure (i)), respectively. Figure (h) ((l), respectively) is the resulting DI of merging the two resulting DIs of applying the run-length smearing both horizontally and vertically on each resulting binarized image of the color layer separation task (*cf.* Figures (f) and (g)) (*cf.* Figures (j) and (k), respectively) by using the logical *OR*.



(a) (b) (c)

Figure 6.7.: Illustration of the second intermediate results of the different tasks performed for homogeneous region extraction from HDIs, using the Gabor features. Figure (a) is the resulting DI of merging the two resulting DIs of applying ARLSA on each resulting binarized DI of the color layer separation task by using the logical OR (*cf.* Figures 6.6(h) and 6.6(l), respectively). Figure (b) shows the resulting DI of labeling the extracted CCs from Figure (a) with taking into consideration the labels of the extracted CCs from Figure 6.6(d) (*i.e.* the refined pixel-labeling results in the *"pixel-labeling refinement"* step by using the majority voting technique). Figure (c) illustrates the output of the proposed algorithm of homogeneous region extraction from HDIs.



(a) (b) (c)

(d) (e) (f) (g)

Figure 6.8.: Illustration of the resulting DIs derived from the proposed algorithm of homogeneous region extraction from HDIs, using the Gabor features. Figure (a) shows an example of HDI (as an input). Figure (b) illustrates the pixel-labeled DI (as an output of the analysis of the extracted Gabor features (graphical regions (blue) and textual regions (green)) (*cf.* Section 5.3.2.2, block 1, Figure 5.1). Figure (c) depicts the outputs of the resulting DI derived from the *"pixel-labeling refinement"* step of the proposed algorithm of homogeneous region extraction from HDIs. Figures (d) and (f) show the resulting DIs derived from the step of extracting and labeling the extracted CCs to identify the homogeneous regions without and with the CC selection task, respectively. Figures (e) and (g) illustrate the resulting DIs derived from the step of homogeneous region extraction without and with the CC selection task, respectively.

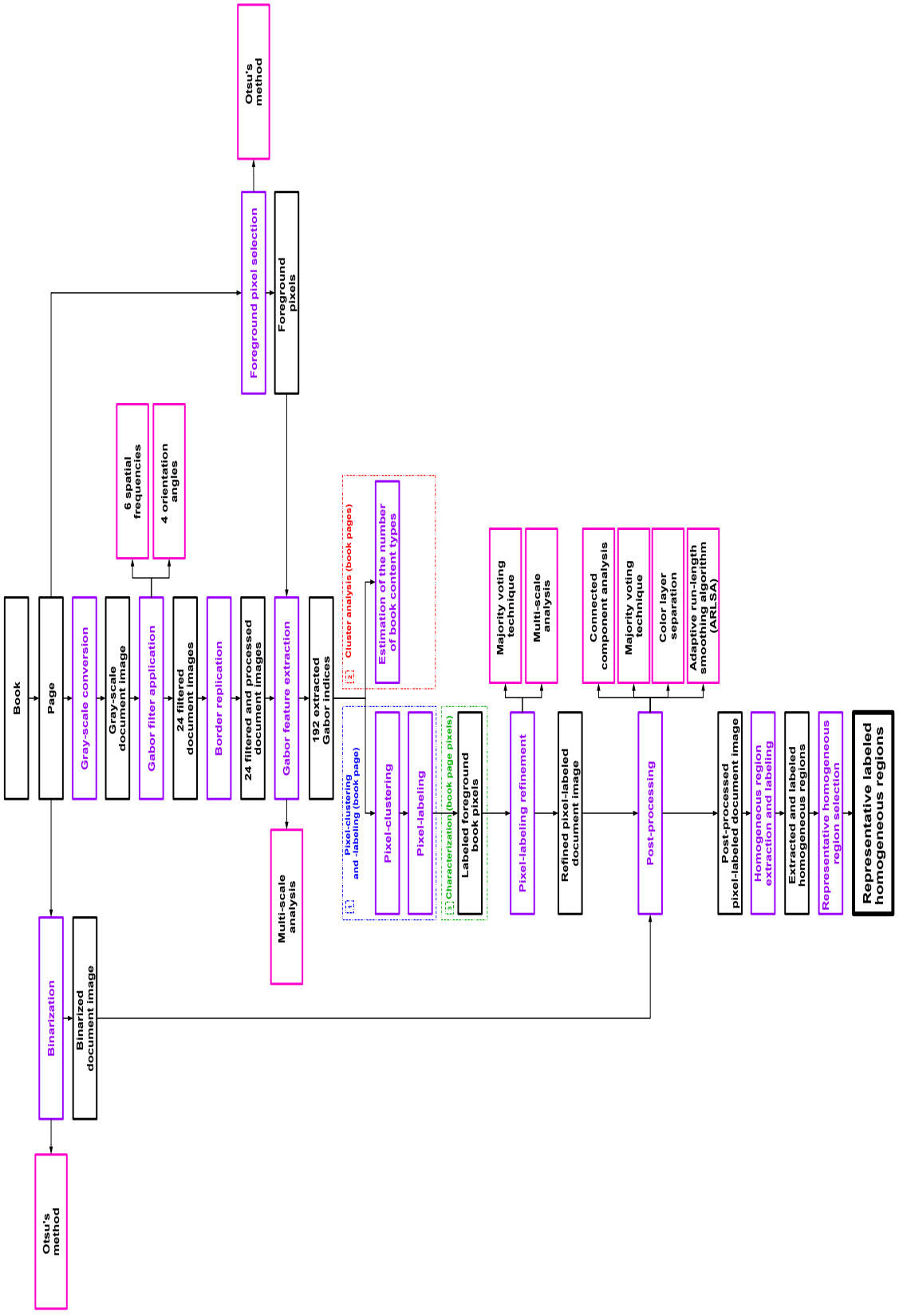


Figure 6.9.: Flowchart of the proposed structural signature for DHB page characterization.

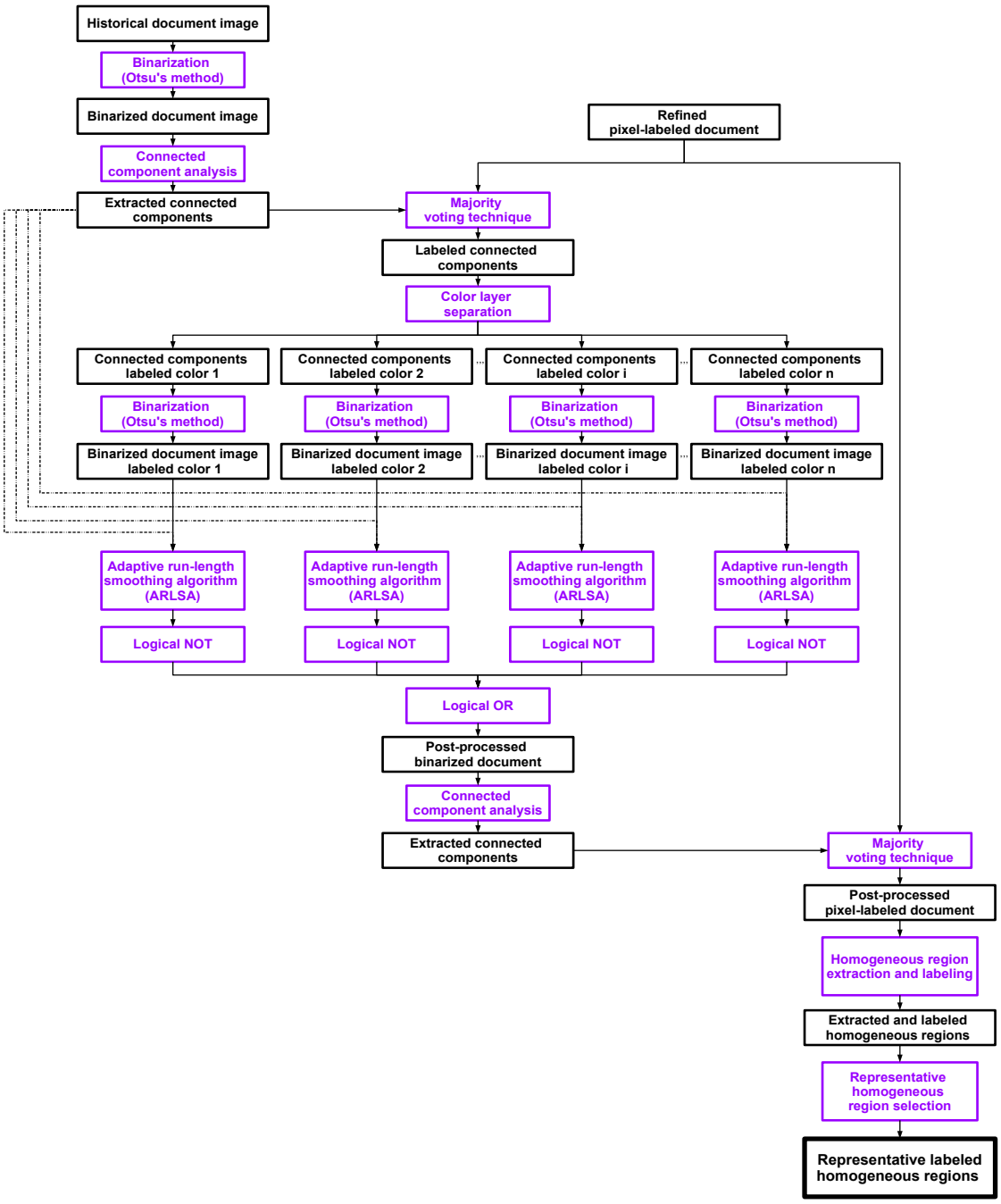


Figure 6.10.: Detailed schematic block representation of the proposed algorithm of homogeneous region extraction step.

6.3.4. Structural signature generation

Since the representative homogeneous regions have been extracted and identified, a signature is needed to define a set of regions of homogeneous texture and their topological relationships. By characterizing each digitized page of ancient book with a set of regions of homogeneous texture and their topological relationships, a signature can be designed for each DHB page. The obtained DHB page signatures help deducing the similarities of DHB page structure or layout and/or content. Indeed, the DHB pages can be compared by categorizing the designed signatures which model the layout and content of DHB pages. Thus, DHB pages with similar layout and/or content can be

grouped. Figure 6.11 provides an example of objectives of the use of a structural signature (*i.e.* finding pages in a DHB which contain similar content component or a group of patterns).

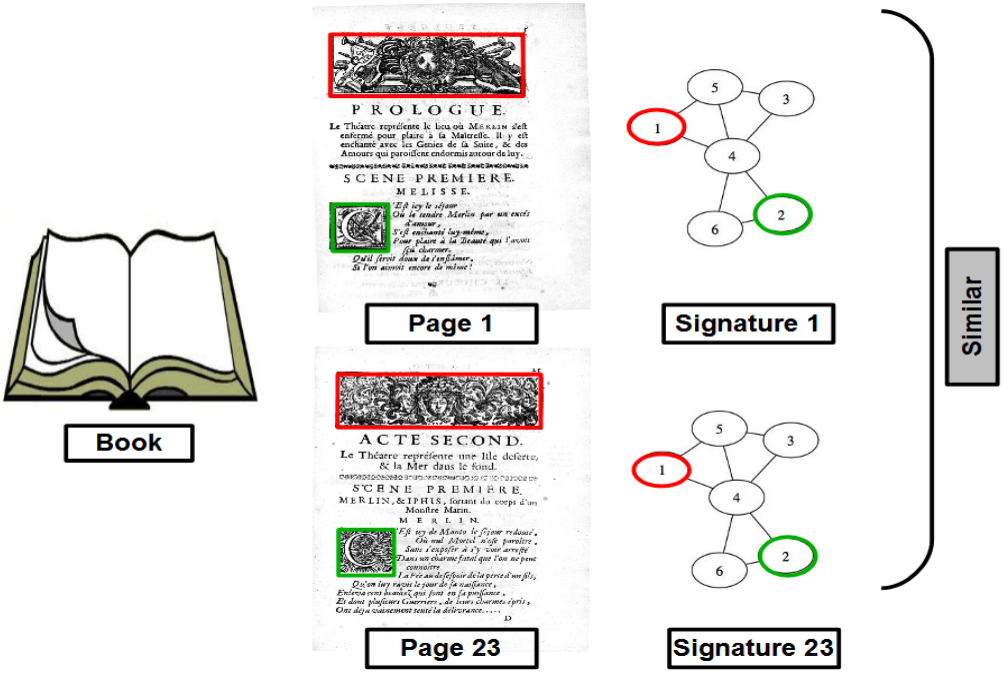


Figure 6.11.: Example of objectives of the use of a structural signature (*i.e.* finding pages in a DHB which contain similar content component or a group of patterns).

Leveraging on the numerous advantages offered by using a structural representation instead a statistical or ontology ones mentioned in Section 6.2.3.2, in this work a structural representation is used in the form of a complete directed attributed graph to generate a page signature (*cf.* Section B.8 for more details about the basic concepts of graphs). For notational convenience complete directed attributed graphs are simply referred to as graphs in the rest of this dissertation. From the extracted homogeneous regions, a complete graph was built, where vertices (G_v) correspond to the extracted homogeneous regions. Each vertex is described by varying low-level information (*i.e.* texture, shape, geometric and topological descriptors). A 238-D feature vector is generated for each vertex, describing and characterizing the extracted homogeneous region (*i.e.* 192 Gabor attributes and 46 shape, geometric and topological descriptors). First, 192 mean Gabor features are retrieved from the extracted Gabor features of the pixels contained in the extracted homogeneous region. In addition to the 192 Gabor features, 46 shape, geometric and topological descriptors are computed from each extracted homogeneous region. Among the computed vertex attributes, several kinds of moments are calculated. The most commonly used moments are the regular (central and normalized central) and Hu moments which have been proposed as features to characterize patterns in classification and recognition applications [468]. Ten spatial moments (m_{ji}), seven central moments (μ_{ji}), seven normalized central moments (ν_{ji}) and seven Hu moments (hu_k) are computed to characterize the shape of the extracted homogeneous regions. In Appendix B and particularly in Section B.7, an exhaustive and detailed review of the different used moment attributes has been carried for generation of structural page representations.

Besides moments used to describe the shape of the extracted regions, geometric and topological descriptors are also computed such as the contour area of the extracted region, topological position of the extracted region centroid in the x-axis, *etc.* The list of the vertex attributes (A^v) is detailed in Table 6.1.

Then, a set of edges (G_e) is built based on topological relationships connecting the different vertices. An edge is built between two vertices, if $F_e^{s,d} \geq Th_e$, where $F_e^{s,d}$ and Th_e denote the edge force and threshold, respectively. The idea of using the edge force ($F_e^{s,d}$) when generating the graph-based signature is to emphasize on the most representative, largest and spatially closest regions. $F_e^{s,d}$ characterizes the gravitation force between two graph vertices: source (G_v^s) and destination (G_v^d). It is deduced from Newton's law of universal gravitation which states that every mass attracts another one by a force pointing along the line intersecting both their centers. The universal gravitation force is directly proportional to the product of the two masses and inversely proportional to the square of the distance between them. However, the edge force ($F_e^{s,d}$) is proportional to the number of pixels of the destination vertex of the built graph (G_v^d), and it is inversely proportional to the square of the Euclidean distance ($ED_{G_v^{s,d}}$) between the two graph vertices: G_v^s and G_v^d. The $F_e^{s,d}$ models the interaction existence and level between two extracted representative homogeneous regions (*i.e.* there is an interaction between two small regions only if they are close to each other, and a large region can have multiple interactions with more distant regions). It is computed as:

$$F_e^{s,d} = \frac{N_{G_v^d}}{(ED_{G_v^{s,d}})^2} \tag{6.1}$$

where $N_{G_v^d}$ denotes the number of pixels of the destination vertex (G_v^d) of the built directed graph. $ED_{G_v^{s,d}}$ denotes the Euclidean distance between the two graph vertices: source (G_v^s) and destination (G_v^d).

Besides the edge force ($F_e^{s,d}$, *cf.* equation 6.1) used to characterize the topological relationships between two extracted regions, two other descriptors are also computed: the absolute differences between the two extracted region centroids in the x and y-axis ($AD_e^{x(s,d)}$ and $AD_e^{y(s,d)}$). The list of the edge attributes (A^e) is detailed in Table 6.1. The edge threshold (Th_e) has been experimentally determined, and it is equal to 0.1. The proposed structural signature is defined according to the algorithm 7.

Table 6.1.: Vertex and edge attributes of a structural signature.

	Id.	Attribute
Vertex	**A- Topological, geometric and shape attributes**	
	A_1^v	Topological position of the extracted region centroid in the x-axis
	A_2^v	Topological position of the extracted region centroid in the y-axis
	A_3^v	Number of pixels of the extracted region
	A_4^v	Contour area of the extracted region
	A_5^v	Contour perimeter of the extracted region
	A_6^v	Topological position of the bounding rectangle of the pixel set of the extracted region in the x-axis
	A_7^v	Topological position of the bounding rectangle of the pixel set of the extracted region in the y-axis
	A_8^v	Height of the bounding rectangle of the pixel set of the extracted region
	A_9^v	Width of the bounding rectangle of the pixel set of the extracted region
	A_{10}^v	Area of the bounding rectangle of the pixel set of the extracted region
	A_{11}^v	Ratio of the height to width of the bounding rectangle of the pixel set of the extracted region
	A_{12}^v	Ratio of the height of the bounding rectangle of the pixel set of the extracted region to the height of the analyzed HDI

Table 6.1 – continued from previous page

	Id.	Attribute
	A^v_{13}	Ratio of the width of the bounding rectangle of the pixel set of the extracted region to the width of the analyzed HDI
	A^v_{14}	Gray-level average of the pixels of the extracted region
	A^v_{15}	Gray-level standard deviation of the pixels of the extracted region
	$A^v_{16\to25}$	10 spatial moments
	$A^v_{26\to32}$	7 central moments
	$A^v_{33\to39}$	7 central normalized moments
	$A^v_{40\to46}$	7 Hu moments
	B- Texture attributes	
	$A^v_{47\to238}$	192 Gabor indices
Edge	A^e_1	Absolute difference between the two extracted region centroids in the x-axis
	A^e_2	Absolute difference between the two extracted region centroids in the y-axis
	A^e_3	Edge force

Figure 6.12 shows the significant role of the selection step of representative homogeneous regions to generate relevant structural signatures for DHB page characterization.

6.4. Experiments and results

To evaluate the performance of the proposed signature-based approach for DHB page characterization, we present in this section qualitative and quantitative evaluation of the different steps of its extraction:

- *"Pixel-labeling refinement"* (*cf.* Sections 6.3.1, B.3 and 6.4.2),
- *"Post-processing"* (*cf.* Sections 6.3.2, B.4 and 6.4.3),
- *"Homogeneous region extraction"* (*cf.* Sections 6.3.3, B.5 and 6.4.4),
- *"Structural signature generation"* (*cf.* Sections 6.3.4, B.6 and 6.4.5).

6.4.1. Experimental corpus and accuracy metrics for performance evaluation

The *"DIGIDOC-Texture dataset"* which is described in Chapter 4 and particularly in Section 4.4.2 is used in this chapter as an experimental corpus. The evaluation using the different accuracy metrics is processed in a similar way to the one used for the experimental evaluation and benchmarking of texture features which has been previously discussed in Chapter 4 and particularly in Section 4.5.1.3. There are two *"Overall"* values, *"Overall*"* and *"Overall**"* in Tables 6.2, 6.3, 6.4, 6.5, 6.6 and 6.7. The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. The *"Two fonts and graphics*"* value represents the case when every font in the text has a distinct label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 3 (graphics and text with two different fonts). The *"Two fonts and graphics**"* value represents the case when all fonts in the text have the same label in the ground-truth and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text). This

Algorithm 7 Generation of a structural signature

$i \leftarrow 1$
while $i \leq N_{HRs}$ **do**
    Compute the 192 mean values of the Gabor indices of the foreground pixels located on i
    Add the 192 mean values of the Gabor indices as vertex attributes
    Compute the 46 shape, geometric and topological indices of i
    Add the 46 shape, geometric and topological indices as vertex attributes
    $i \leftarrow i + 1$
$s \leftarrow 1$
while $s \leq N_{HRs}$ **do**
    $d \leftarrow s + 1$
    while $d \leq N_{HRs}$ **do**
        if $F_e^{s,d} \geq Th_e$ **then**
            Define an edge between s and d
            Add the value of $F_e^{s,d}$ as the first edge attribute
            Compute $AD_e^{x(s,d)}$
            Add the value of $AD_e^{x(s,d)}$ as the second edge attribute
            Compute $AD_e^{y(s,d)}$
            Add the value of $AD_e^{y(s,d)}$ as the third edge attribute
        $d \leftarrow d + 1$
    $s \leftarrow s + 1$

where N_{HRs} denotes the number of the extracted homogeneous regions or graph vertices. $AD_e^{x(s,d)}$ and $AD_e^{y(s,d)}$ denote the absolute difference between the two extracted region centroids (s and d) in the x and y-axis, respectively.

distribution points out which texture features can be more adequate for segmenting documents containing two text fonts and graphics into two/three classes, *i.e.* separating two distinct text fonts when the documents contain graphics.

First, to evaluate quantitatively the different obtained results of the *"Pixel-labeling refinement"* and *"Post-processing"* steps, the following clustering and classification accuracy measures (J, PPB, P, R, F and CA) which have been previously detailed in Chapter 4 and particularly in Section 4.4.3.

On the other side, three per-pixel accuracy metrics, the area precision (P_{AR}), area recall (R_{AR}) and Jaccard index (J_{AR}), are computed for evaluating the extracted homogeneous regions [469]. Assume the number of foreground pixels defined in the area i of the bounding box of the result block is $|B_r^i|$ and the number of foreground pixels defined in the area i of the bounding box of the ground-truth is $|B_{gt}^i|$.

- The ***area precision*** computes the overlaying ratio of the number of foreground pixels defined in the area of B_r^i by the one defined in the area of B_{gt}^i. It is given by:

$$P_{AR}^i(B_r^i, B_{gt}^i) = \frac{|B_r^i \cap B_{gt}^i|}{|B_r^i|} \tag{6.2}$$

- The ***area recall*** calculates the covering ratio of the number of foreground pixels defined in the area of B_{gt}^i by the one defined in the area of B_r^i. It is given by:

$$R_{AR}^i(B_r^i, B_{gt}^i) = \frac{|B_r^i \cap B_{gt}^i|}{|B_{gt}^i|} \tag{6.3}$$

- The ***Jaccard index*** measures the overlap ratio between the number of foreground pixels defined in the two areas B_r^i and B_{gt}^i. It is given by:

$$J_{AR}^i(B_r^i, B_{gt}^i) = \frac{|B_r^i \cap B_{gt}^i|}{|B_r^i \cup B_{gt}^i|} \tag{6.4}$$

6.4.2. Pixel-labeling refinement

In this section, qualitative and quantitative results are given to illustrate the potential to introduce the *"Pixel-labeling refinement"* step into the auto-correlation and Gabor-based pixel-labeling schemes. Figures 6.13, 6.14, B.26, B.27 and B.28 illustrate the qualitative results of introducing the *"Pixel-labeling refinement"* step into the auto-correlation and Gabor-based pixel-labeling schemes, in *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* HDIs from the *"DIGIDOC-Texture dataset"*, respectively.

In Figure B.27 illustrating an *"One font and graphics"* HDI from the *"DIGIDOC-Texture dataset"*, we note that when comparing the two results of the *"Pixel-labeling"* step using that the auto-correlation and Gabor features without introducing the *"Pixel-labeling refinement"* step, the auto-correlation-based approach performs better than the Gabor one in discriminating graphic (green) and text (blue) regions (*cf.* Figure B.27(a)). The Gabor features have more difficulty separating textual regions (blue) when they are too spatially close to the graphical ones (*i.e.* textual regions which are spatially close to the graphic ones have been mis-labeled) (*cf.* Figure B.27(c)). Furthermore, by comparing visually the auto-correlation and Gabor-based pixel-labeling results of introducing the *"Pixel-labeling refinement"* step into the texture-based pixel-labeling scheme, illustrated in HDIs from the *"DIGIDOC-Texture dataset"*, we observe that HDI content regions are visibly becoming more homogeneous especially when using the auto-correlation features (*cf.* Figures 6.13(b) and 6.14(b)). However, we note that the Gabor feature analysis does not require the *"Pixel-labeling refinement"* step, since there is a no visually improvement difference between the cases of without and with the *"Pixel-labeling refinement"* step (*cf.* Figures 6.13(d) and 6.14(d)). This confirms that introducing the spatial or topographical relationship between pixels by using the spatial multi-scale analysis of majority votes into the texture-based pixel-labeling scheme can improve significantly the performance depending on the quality of the initial pixel-labeling results (*i.e.* without taking into consideration the topographical relationships of pixels and their labels). Nevertheless, we note that the mis-labeling errors of the pixel-labeling produced in Figure 6.13(c) have not been rectified by introducing the spatial or topographical relationship between pixels by means of the spatial multi-scale analysis of majority votes into the texture-based pixel-labeling scheme Figure (*cf.* Figure 6.13(d)). This is due to the inherent pixel-labeling errors in the *"Pixel-labeling"* step produced when only analyzing the auto-correlation features. These errors can be avoided if the texture and topographical features are analyzed simultaneously (*cf.* Figure 6.13(c)). Other qualitative results are given to demonstrate the performance of the *"Pixel-labeling refinement"* step in Appendix B and particularly in Section B.3.

To demonstrate the robustness of the *"Pixel-labeling refinement"* step and provide additional insights into its classification accuracy, numerous clustering accuracy metrics and classification accuracy rates (J, PPB, P, R, F and CA) are computed. Table 6.2 presents the quantitative assessment of the *"pixel-labeling refinement"* step using the results of the auto-correlation and Gabor-based pixel-labeling schemes with the *"DIGIDOC-Texture dataset"*. Table 6.3 presents the difference values in the computed clustering and classification accuracy measures when introducing the *"Pixel-labeling refinement"* step and without it into the auto-correlation and Gabor-based pixel-labeling schemes using the *"DIGIDOC-Texture dataset"*.

We observe that the two best average performances for most of the computed evaluation metrics are obtained for the *"One font and graphics"* and *"Two fonts and graphics**"* categories of the *"DIGIDOC-Texture dataset"* with using the auto-correlation (95%(PPB), 86%(P), 86%(R), 85%(F) and 91%(CA) for the *"One font and graphics"* HDI category, and 94%(PPB), 87%(P),

86%(R), 86%(F) and 87%(CA) for the *"Two fonts and graphics**"*) and Gabor (96%(PPB), 90%(P), 86%(R), 88%(F) and 88%(CA) for the *"One font and graphics"* HDI category, and 98%(PPB), 91%(P), 88%(R), 89%(F) and 89%(CA) for the *"Two fonts and graphics**"*) features. On the other side, we observe that the worst average performances for most of the computed evaluation metrics are obtained for the *"Only three fonts"* category of the *"DIGIDOC-Texture dataset"* with using the auto-correlation (87%(PPB), 58%(P), 63%(R), 60%(F) and 74%(CA)) and Gabor (88%(PPB), 67%(P), 62%(R), 64%(F) and 68%(CA)) features. As a consequence, we note that the ranking of the different categories of the *"DIGIDOC-Texture dataset"* obtained when introducing the *"Pixel-labeling refinement"* step into the auto-correlation and Gabor-based pixel-labeling schemes is similar to the one obtained without the *"Pixel-labeling refinement"* step (*i.e.* without taking into consideration the topographical relationships of pixels and their labels). We observe that the overall average performances by the auto-correlation (79%(J), 91%(PPB), 71%(P), 73%(R), 71%(F) and 82%(CA) for *"Overall*"*, and 80%(J), 92%(PPB), 76%(P), 78%(R), 77%(F) and 88%(CA) for *"Overall**"*) and Gabor (75%(J), 93%(PPB), 79%(P), 74%(R), 76%(F) and 79%(CA) for *"Overall*"*, and 78%(J), 94%(PPB), 84%(P), 79%(R), 81%(F) and 82%(CA) for *"Overall**"*) features.

Furthermore, we conclude from Table 6.3 that we have a significant overall gain in performance when introducing the *"Pixel-labeling refinement"* step into the auto-correlation-based pixel-labeling scheme (gains of 11.1%(J), 7.3%(PPB), 1.3%(P), 4.6%(R), 2.8%(F) and 8%(CA) for *"Overall*"*, and 9.8%(J), 6.3%(PPB), 0.9%(P), 4.1%(R), 2.4%(F) and 6.4%(CA) for *"Overall**"*). On the other side, there is a no significant gain in the case of introducing the *"Pixel-labeling refinement"* step into the Gabor-based pixel-labeling scheme. The quantitative assessment strengthens our previous visual observations that introducing the spatial or topographical relationship between pixels by using the spatial multi-scale analysis of majority votes into the texture-based pixel-labeling scheme can improve significantly the performance depending on the quality of the initial pixel-labeling results (*i.e.* without taking into consideration the topographical relationships of pixels and their labels).

6.4.3. Post-processing

To show the potential to introduce the *"Post-processing"* step after the *"Pixel-labeling refinement"* task, into the auto-correlation and Gabor-based pixel-labeling schemes, the performance of the *"Post-processing"* step has been discussed in this section. Figures 6.15, 6.16, B.29, B.30 and B.31 illustrate the qualitative results of introducing the *"Post-processing"* step after the *"Pixel-labeling refinement"* task, into the auto-correlation and Gabor-based pixel-labeling schemes, in *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* HDIs from the *"DIGIDOC-Texture dataset"*, respectively.

By comparing visually the two results of introducing the *"Post-processing"* step after the *"Pixel-labeling refinement"* task, into the auto-correlation and Gabor-based pixel-labeling schemes, illustrated in HDIs from the *"DIGIDOC-Texture dataset"*, we observe that HDI content regions are visibly becoming more homogeneous for both the auto-correlation and Gabor features (*cf.* Figures 6.15(b), 6.16(b), 6.15(d) and 6.16(d)). Furthermore, we note that based on the Gabor features in the texture-based pixel-labeling scheme, the *"Post-processing"* step ensure the discrimination between graphic and text regions (*cf.* Figures 6.16(d) and B.29(d)) and the segmentation of different text fonts (*cf.* Figures B.30(d) and B.31(d)). We observe that by means of the *"Post-processing"* step the pixel-labeling results are good, *i.e.* there is no mis-labeled pixels, and pure and homogeneous regions are generated. Nevertheless, we note that the mis-labeling errors of the pixel-labeling produced in Figure 6.15(c) have not been rectified by introducing the *"Post-processing"* step into the texture-based pixel-labeling scheme Figure (*cf.* Figure 6.15(d)). This also due to the inherent pixel-labeling errors in the *"Pixel-labeling"* step produced when only analyzing the auto-correlation features (*cf.* Figure 6.13(c)). Other qualitative results are given to demonstrate the performance of the *"Post-processing"* step in Appendix B and particularly in Section B.4.

To demonstrate the robustness of the *"Post-processing"* step and provide additional insights into its classification accuracy, numerous clustering accuracy metrics and classification accuracy rates (J, PPB, P, R, F and CA) are computed. Table 6.4 presents the quantitative assessment of the *"Post-processing"* step using the results of the *"Pixel-labeling refinement"* task performed on the auto-correlation and Gabor-based pixel-labeling schemes with the *"DIGIDOC-Texture dataset"*. Table 6.5 presents the difference values in the computed clustering and classification accuracy measures when introducing the *"Post-processing"* step and without it on the results of the *"Pixel-labeling refinement"* task into the auto-correlation and Gabor-based pixel-labeling schemes using the *"DIGIDOC-Texture dataset"*.

We observe that the two best average performances for most of the computed evaluation metrics are obtained for the *"One font and graphics"* and *"Two fonts and graphics**"* categories of the *"DIGIDOC-Texture dataset"* with using the auto-correlation (99%(PPB), 78%(P), 75%(R), 76%(F) and 92%(CA) for the *"One font and graphics"* HDI category, and 99%(PPB), 85%(P), 83%(R), 84%(F) and 91%(CA) for the *"Two fonts and graphics**"*) and Gabor (96%(PPB), 82%(P), 80%(R), 81%(F) and 93%(CA) for the *"One font and graphics"* HDI category, and 99%(PPB), 90%(P), 88%(R), 89%(F) and 89%(CA) for the *"Two fonts and graphics**"*) features. On the other side, we observe that the worst average performances for most of the computed evaluation metrics are obtained for the *"Only three fonts"* category of the *"DIGIDOC-Texture dataset"* with using the auto-correlation (92%(PPB), 56%(P), 54%(R), 54%(F) and 78%(CA)) and Gabor (89%(PPB), 64%(P), 62%(R), 62%(F) and 71%(CA)) features. As a consequence, we note that the ranking of the different categories of the *"DIGIDOC-Texture dataset"* obtained when introducing the *"Post-processing"* step using the results of the *"Pixel-labeling refinement"* task into the auto-correlation and Gabor-based pixel-labeling schemes is similar to the one obtained without the *"Post-processing"* step. We observe that the overall average performances by the auto-correlation (85%(J), 96%(PPB), 69%(P), 67%(R), 67%(F) and 87%(CA) for *"Overall*"*, and 85%(J), 97%(PPB), 73%(P), 71%(R), 72%(F) and 87%(CA) for *"Overall**"*) and Gabor (80%(J), 94%(PPB), 73%(P), 71%(R), 72%(F) and 83%(CA) for *"Overall*"*, and 81%(J), 95%(PPB), 79%(P), 77%(R), 78%(F) and 85%(CA) for *"Overall**"*) features.

Furthermore, we conclude from Table 6.5 that we have a slight gain in computing the J and PPB accuracy metrics, while a slight drop in calculating the P, R, F and CA evaluation measures when introducing the *"Post-processing"* step after the *"Pixel-labeling refinement"* task into the auto-correlation and Gabor-based pixel-labeling schemes. This slight variability when computing the different evaluation metrics can be explained that using rectangles in zoning the ground-truth can affect some per-pixel accuracy metrics for a quantitative assessment. Indeed, the background pixels are retained with the foreground ones when introducing the *"Post-processing"* step which is based on filling automatically the space within each CC.

6.4.4. Homogeneous region extraction

This section presents an assessment of the *"Homogeneous region extraction"* step, performed after the *"Post-processing"* task on the auto-correlation and Gabor-based pixel-labeling schemes. Figures 6.17, 6.18, B.32, B.33 and B.34 illustrate the qualitative results of the *"Homogeneous region extraction"* step, performed after the *"Post-processing"* task on the auto-correlation and Gabor-based pixel-labeling schemes, in *"One font and graphics"*, *"Two fonts and graphics*"*, *"Two fonts and graphics**"*, *"Only two fonts"* and *"Only three fonts"* HDIs from the *"DIGIDOC-Texture dataset"*, respectively.

We observe that homogeneous regions are correctly extracted depending on the results of the *"Post-processing"* task, by comparing visually the results of the *"Homogeneous region extraction"* step, performed after the *"Post-processing"* task on the auto-correlation and Gabor-based pixel-labeling schemes, illustrated in HDIs from the *"DIGIDOC-Texture dataset"*. For instance, in Figures 6.17(b) and 6.17(d), a green bounding box is drawn (*i.e.* green color attributed to the drawn bounding box represents a graphic regions) to cover all the pixels belonging to the extracted CC

which corresponds to the noise information on the borders of the analyzed HDI. On the other side, for an *"One font and graphics"* HDI, we note that the *"Homogeneous region extraction"* step gives satisfying results, since we exactly extracted and distinguish two representative regions with different contents, graphic (green) and text (red) regions (*cf.* Figures 6.18(b) and 6.18(d)). Other qualitative results are given to demonstrate the performance of the *"Homogeneous region extraction"* step in Appendix B and particularly in Section B.5.

Afterwards, three accuracy metrics (P_{AR}, R_{AR} and J_{AR}) are computed to evaluate the performance of the *"Homogeneous region extraction"* step. Table 6.6 presents quantitative assessment of the *"Homogeneous region extraction"* step performed after the *"Post-processing"* task on the auto-correlation and Gabor-based pixel-labeling schemes using the *"DIGIDOC-Texture dataset"*. Table 6.7 presents the difference values in the computed accuracy metrics for the evaluation of the *"Homogeneous region extraction"* step performed after the *"Post-processing"* task between the auto-correlation and Gabor-based pixel-labeling schemes using the *"DIGIDOC-Texture dataset"*. The computed accuracy metrics values are quite encouraging since we observe that the overall average performances by the auto-correlation (94%(P_{AR}), 81%(R_{AR}) and 78%(J_{AR}) for *"Overall*"*, and 94%(P_{AR}), 80%(R_{AR}) and 77%(J_{AR}) for *"Overall**"*) and Gabor (94%(P_{AR}), 80%(R_{AR}) and 78%(J_{AR}) for *"Overall*"*, and 94%(P_{AR}), 80%(R_{AR}) and 77%(J_{AR})) features. Hence, we note that we have a higher precision and a lower recall when extracting homogeneous regions from HDIs. Therefore, we conclude that the proposed method gives satisfying and perfectible results in extracting correctly homogeneous regions from HDIs. However, there is at least 20% of foreground pixels which are not retrieved by our algorithm of homogeneous region extraction. This can be justified by the selection step of representative homogeneous regions used to generate relevant structural signatures for HDI characterization.

Similar to the previous steps (*"Pixel-labeling refinement"* and *"Post-processing"*) of the proposed method used to generate a structural signature for DHB page characterization (Section 6.4.2 and 6.4.3), we observe that the two best average performances for most of the computed evaluation metrics are obtained for the *"One font and graphics"* and *"Only two fonts"* categories of the *"DIGIDOC-Texture dataset"* with using the auto-correlation 96%(P_{AR}), 90%(R_{AR}) and 88%(J_{AR}) for the *"One font and graphics"* HDI category, and 97%(P_{AR}), 80%(R_{AR}) and 79%(J_{AR}) for the *"Only two fonts"*) and Gabor (97%(P_{AR}), 90%(R_{AR}) and 88%(J_{AR}) for the *"One font and graphics"* HDI category, and 97%(P_{AR}), 79%(R_{AR}) and 78%(J_{AR}) for the *"Only two fonts"*) features. On the other side, we observe that the worst average performances for most of the computed evaluation metrics are obtained for the *"Only three fonts"* category of the *"DIGIDOC-Texture dataset"* with using the auto-correlation (91%(P_{AR}), 73%(R_{AR}) and 69%(J_{AR})) and Gabor (92%(P_{AR}), 73%(R_{AR}) and 68%(J_{AR})) features. We can state that the *"Pixel-labeling"* step constitutes a key task toward having good performance for HDI segmentation and characterization.

Furthermore, we conclude from Table 6.7 that by comparing several evaluation accuracy metrics to assess the *"Homogeneous region extraction"* step performed after the *"Post-processing"* task between the auto-correlation and Gabor-based pixel-labeling schemes using the *"DIGIDOC-Texture dataset"*, we have no a significant difference between the two Gabor and auto-correlation-based approaches (P_{AR} difference values of 0.4% and 0.6% for *"Overall*"* and *"Overall**"*). This straightens our previous observation concerning the influence of using rectangles in zoning the ground-truth to compute several per-pixel accuracy metrics for a quantitative assessment of the *"Homogeneous region extraction"* step. Moreover, the use of a selection step of representative homogeneous regions which is performed to generate relevant structural signatures for HDI characterization can have an impact in assessing the performance of the *"Homogeneous region extraction"* step.

6.4.5. Structural signature generation

Figures 6.19, 6.20, B.35, B.36 and B.37 illustrate the qualitative results of the *"Structural signature generation"* step, performed after the *"Homogeneous region extraction"* task on the auto-correlation and Gabor-based pixel-labeling schemes, in *"One font and graphics"*, *"Two fonts and graphics*"*,

*"Two fonts and graphics** "*, *"Only two fonts"* and *"Only three fonts"* HDIs from the *"DIGIDOC-Texture dataset"*, respectively. In Figures 6.19 and 6.20, we can see different oriented arrows are drawn to model the interaction existence between two extracted representative homogeneous regions. In Appendix B and particularly in Section B.6, other visual results of the *"Structural signature generation"* step, performed after the *"Homogeneous region extraction"* task on the auto-correlation and Gabor-based pixel-labeling schemes are illustrated.

The quantitative assessment of the *"Structural signature generation"* step will be among the next chapter to illustrate the effectiveness of the proposed page signature (*cf.* Chapter 7).

6.5. Discussion

An automatic characterization method of DHB pages is proposed in this chapter. However, the performance of the proposed method is strongly dependent on the quality of the initial pixel-labeling results due to the pipeline/building-block of the proposed automatic characterization approach of DHB pages.

To illustrate the performance of the proposed method for DHB page characterization by means of a structural representation, a detailed experimental evaluation has been conducted through a quantitative assessment of the different steps of the proposed approach used is presented. Nevertheless, the fundamental question is if the proposed method has been assessed properly or not. Indeed, we observe a slight variability when computing the different evaluation metrics. As a result, we can state that the influence of using rectangles in zoning the ground-truth to compute several per-pixel accuracy metrics is significant for a thorough quantitative assessment of the different steps of the proposed method. Baird *et al.* pointed out the zoning methodology problems and reported three accuracy metrics (per-pixel accuracy, per-page inventory accuracy and subjective segmentation quality) for a pixel-based approach evaluation [470, 471, 316]. They reported that using rectangles in zoning can affect the per-pixel accuracy score due to the fact that some content can not be described by rectangular zones (e.g. handwritten regions) and due to the arbitrariness and inconsistency in zoning. They also noted that using rectangles in zoning has an influence to compute the per-page inventory accuracy since the information of page layout is not included. This confirms questions about the defined ground-truth which is to a certain extent subjective and it is difficult to acquire a pixel-based ground-truth. Further work is needed to solve this issue even it is worth noting that it is a straightforward task to define a pixel-based ground-truth. Our future work will focusing on analyzing four other state-of-the-art ground-truthing tools, TrueViz[1], WebGT[2], Aletheia[3] and Divadia[71] for more reliable performance evaluations.

6.6. Conclusion

Throughout this chapter, a description of an automatic characterization approach of DHB pages is proposed. The DHB page characterization is based on texture, shape, geometry and topographical descriptors. The characterization is embedded in what we call a structural signature of DI. Generating a structural signature for each analyzed DHB page is carried out in three stages. The first step consists in refining the obtained pixel-labeling results by taking into account the topological or spatial relationships between pixels. The second one aims to extract homogeneous regions. Finally, the third one is generating a structural signature of the page layout and content.

The extraction of homogeneous regions is based on texture features, multi-scale analysis, an ARLSA, CC analysis technique and majority voting approach. Having extracted homogeneous regions, the topological relationships between regions in each page are used to construct a texture-based structural signature in the form of a graph. The obtained signature defines both the spatial

[1] `http://www.kanungo.com/software/software.html#trueviz`
[2] `http://win-web.cs.bgu.ac.il/`
[3] `http://www.primaresearch.org/tools`

organization of the extracted homogeneous texture regions and the different attributes that characterize those regions. The proposed characterization approach of DHB pages gives encouraging results since 77% of Jaccard index is noted when we have evaluated the extracted homogeneous regions.

The proposed DHB page signature extraction process is independent of the layout and content of the analyzed DHB pages, and hence, it is applicable to a large variety of HDIs. Indeed, it does not assume *a priori* knowledge regarding page content and structure.

Supported by the fact that the proposed page signature provides a topological signature of a DHB page under consideration characterizing mainly the layout structure and/or typographic/graphical characteristics of the HDI content, several signature-based applications for managing effectively a corpus or collections of DHBs or HDIs should be implemented. These applications will illustrate the potential of the proposed page signature to index, gather, compare, categorize or group DHB pages according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the HDI content. The assessment of few potential applications of the proposed DHB page signature is presented in the next chapter (*cf.* Chapter 7).

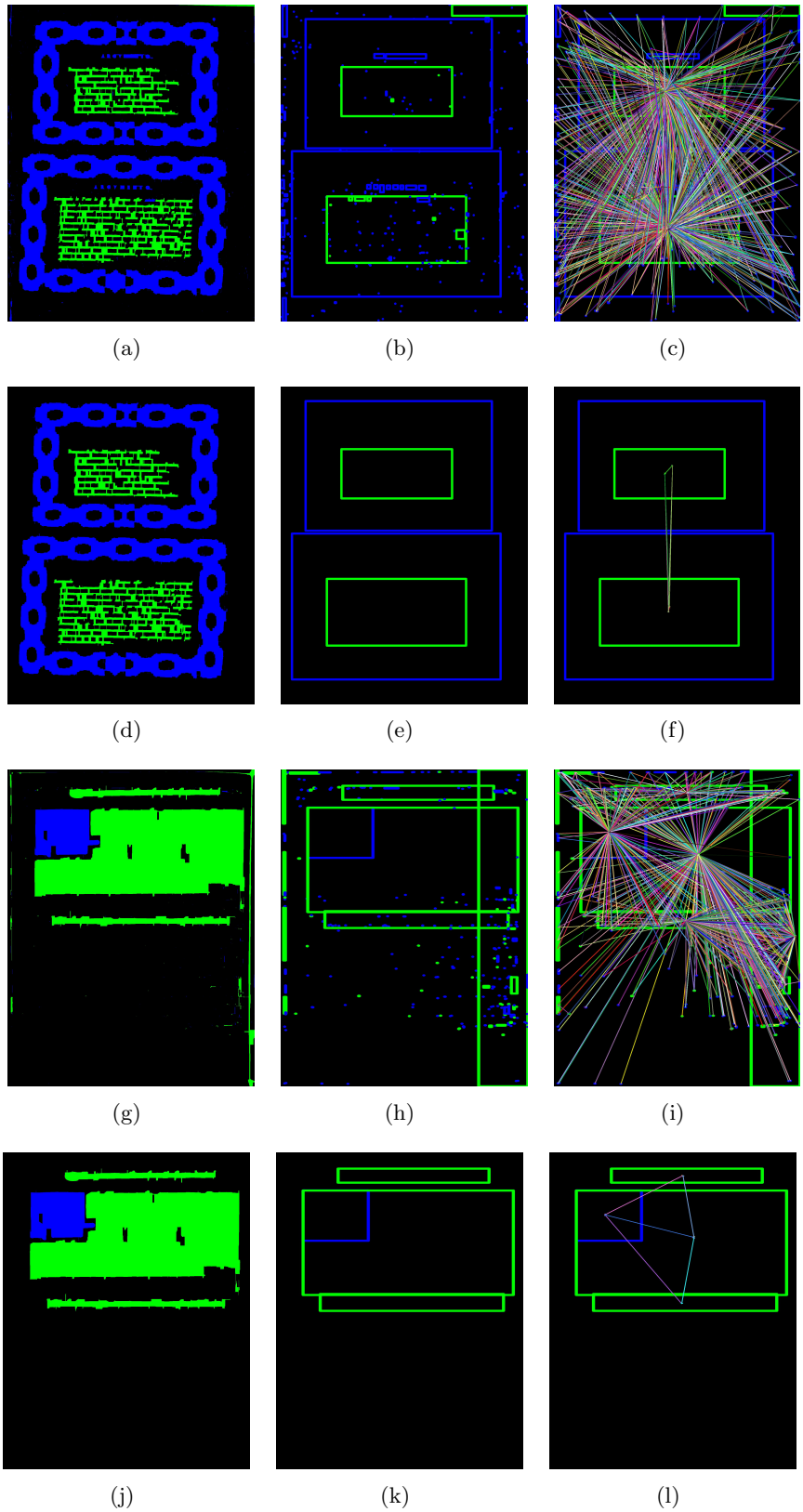


Figure 6.12.: Illustration of two examples of structural signatures for DHB page characterization. Figures (a,d) and (g,j) show the resulting DIs derived from the step of extracting and labeling the extracted CCs to identify the homogeneous regions without and with the CC selection task, respectively. Figures (b,e) and (h,k) illustrate the resulting DIs derived from the step of homogeneous region extraction without and with the CC selection task, respectively. Figures (c,f) and (i,l) show the generated structural signatures for DHB page characterization without and with the CC selection task, respectively.

Auto-correlation

"Pixel-labeling"

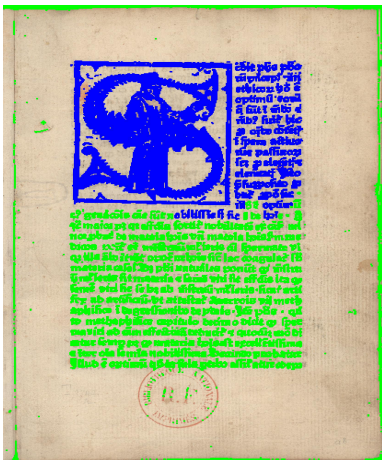
(a) $PPB = 0.95$ $CA = 0.93$
$P = 0.92$ $R = 0.93$ $F = 0.93$

+ *"Pixel-labeling refinement"*

(b) $PPB = 0.97$ $CA = 0.95$
$P = 0.94$ $R = 0.96$ $F = 0.95$

Gabor

"Pixel-labeling"



(c) $PPB = 0.99$ $CA = 0.87$
$P = 0.88$ $R = 0.88$ $F = 0.88$

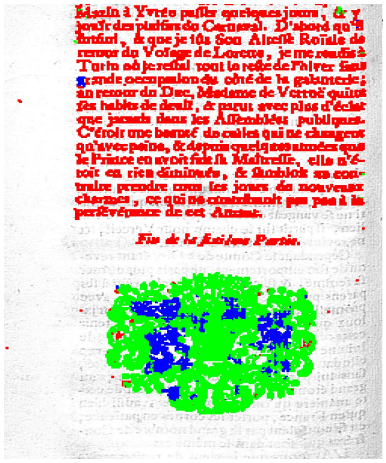
+ *"Pixel-labeling refinement"*

(d) $PPB = 0.99$ $CA = 0.86$
$P = 0.88$ $R = 0.88$ $F = 0.88$

Figure 6.13.: Examples of introducing the ***"Pixel-labeling refinement"*** step into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"One font and graphics"*** HDI from the *"DIGIDOC-Texture dataset"*.

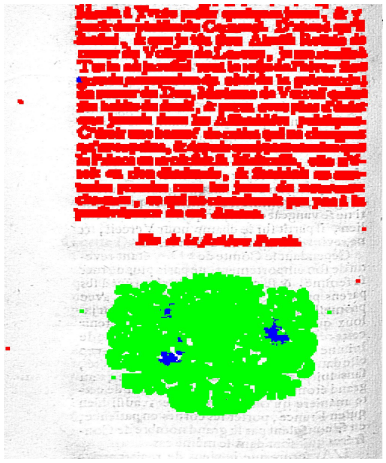
Auto-correlation

"Pixel-labeling"



(a) $PPB = 0.93$ $CA = 0.88$
$P = 0.60$ $R = 0.65$ $F = 0.62$

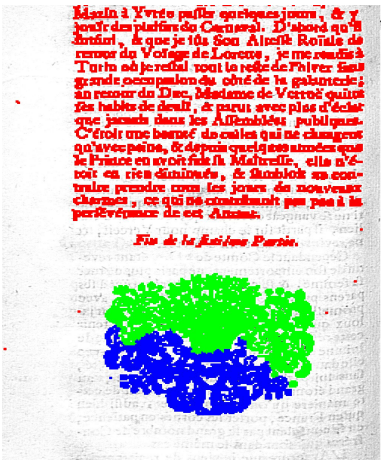
+ *"Pixel-labeling refinement"*



(b) $PPB = 0.99$ $CA = 0.97$
$P = 0.66$ $R = 0.66$ $F = 0.66$

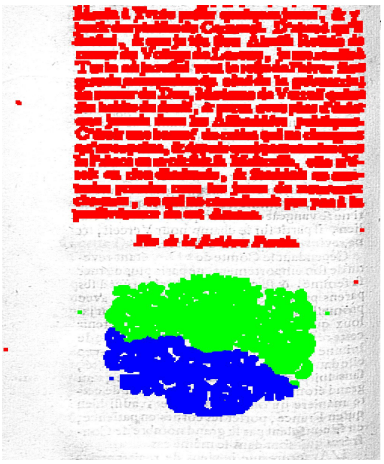
Gabor

"Pixel-labeling"



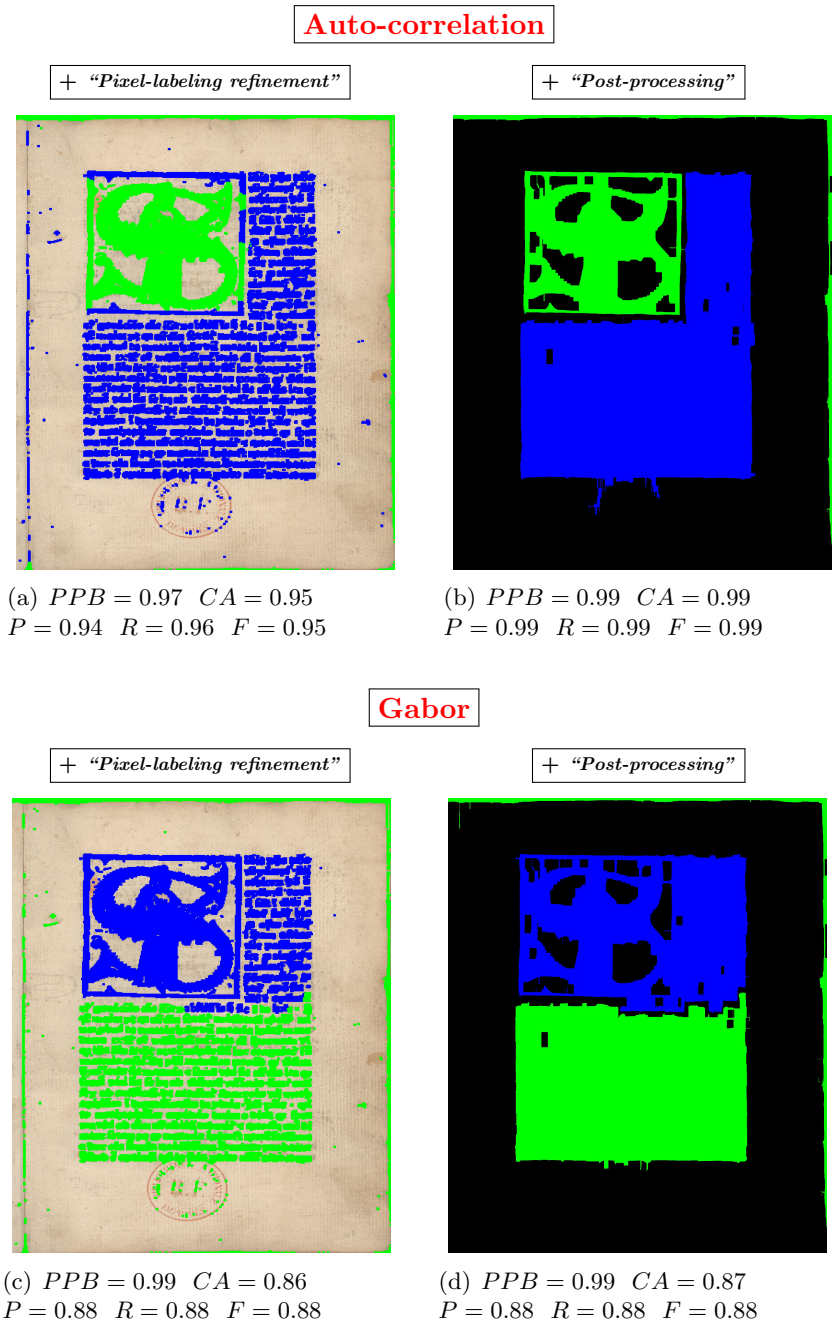
(c) $PPB = 0.86$ $CA = 0.77$
$P = 0.53$ $R = 0.66$ $F = 0.58$

+ *"Pixel-labeling refinement"*



(d) $PPB = 0.86$ $CA = 0.78$
$P = 0.53$ $R = 0.66$ $F = 0.59$

Figure 6.14.: Examples of introducing the ***"Pixel-labeling refinement"*** step into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics*"*** HDI from the *"DIGIDOC-Texture dataset"*.



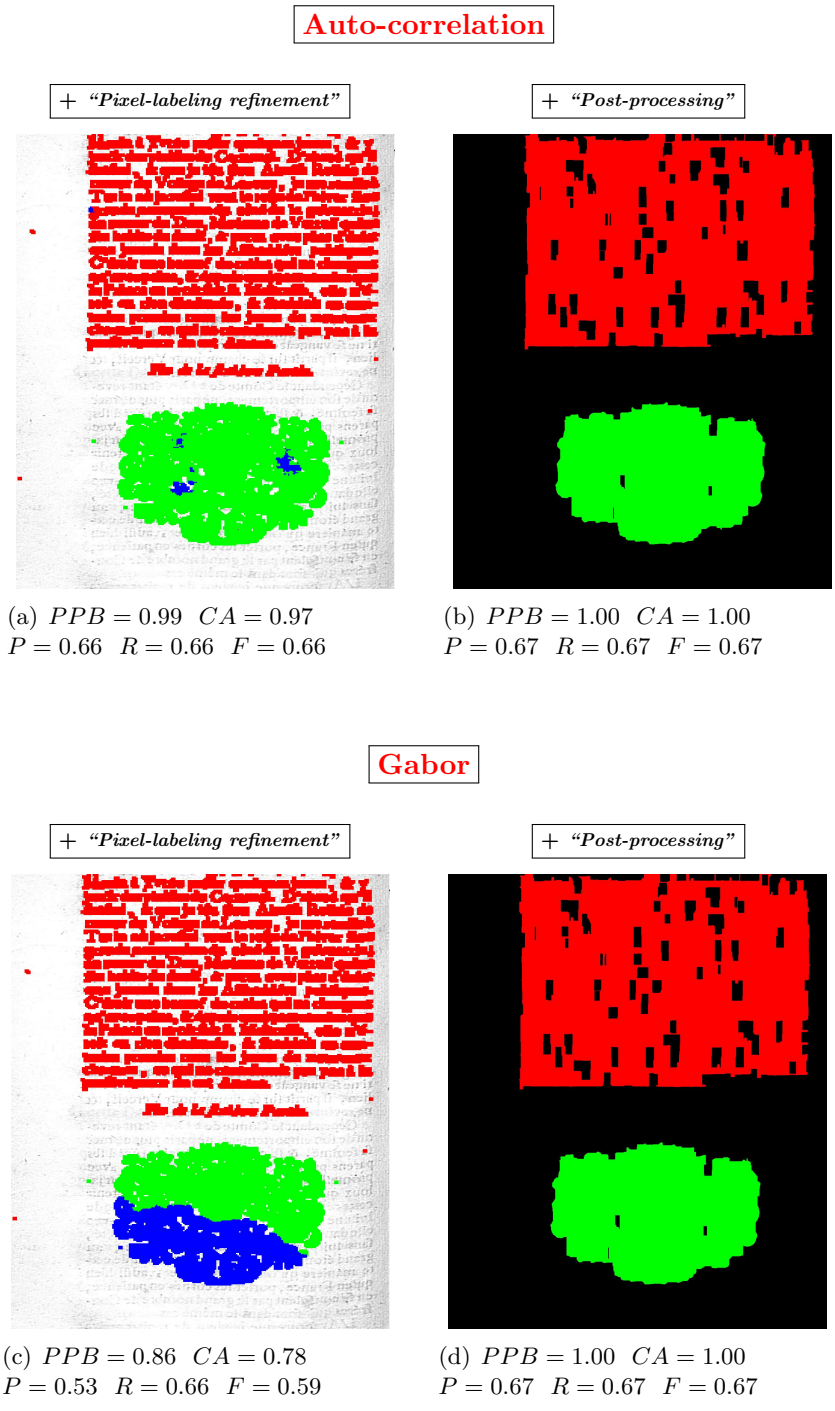
(a) $PPB = 0.97$ $CA = 0.95$
$P = 0.94$ $R = 0.96$ $F = 0.95$

(b) $PPB = 0.99$ $CA = 0.99$
$P = 0.99$ $R = 0.99$ $F = 0.99$

(c) $PPB = 0.99$ $CA = 0.86$
$P = 0.88$ $R = 0.88$ $F = 0.88$

(d) $PPB = 0.99$ $CA = 0.87$
$P = 0.88$ $R = 0.88$ $F = 0.88$

Figure 6.15.: Examples of introducing the ***"Post-processing"*** step after the *"Pixel-labeling refinement"* task, into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"One font and graphics"*** HDI from the *"DIGIDOC-Texture dataset"*.



(a) $PPB = 0.99$ $CA = 0.97$
$P = 0.66$ $R = 0.66$ $F = 0.66$

(b) $PPB = 1.00$ $CA = 1.00$
$P = 0.67$ $R = 0.67$ $F = 0.67$

(c) $PPB = 0.86$ $CA = 0.78$
$P = 0.53$ $R = 0.66$ $F = 0.59$

(d) $PPB = 1.00$ $CA = 1.00$
$P = 0.67$ $R = 0.67$ $F = 0.67$

Figure 6.16.: Examples of introducing the ***"Post-processing"*** after the *"Pixel-labeling refinement"* task, into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics**"*** HDI from the *"DIGIDOC-Texture dataset"*.

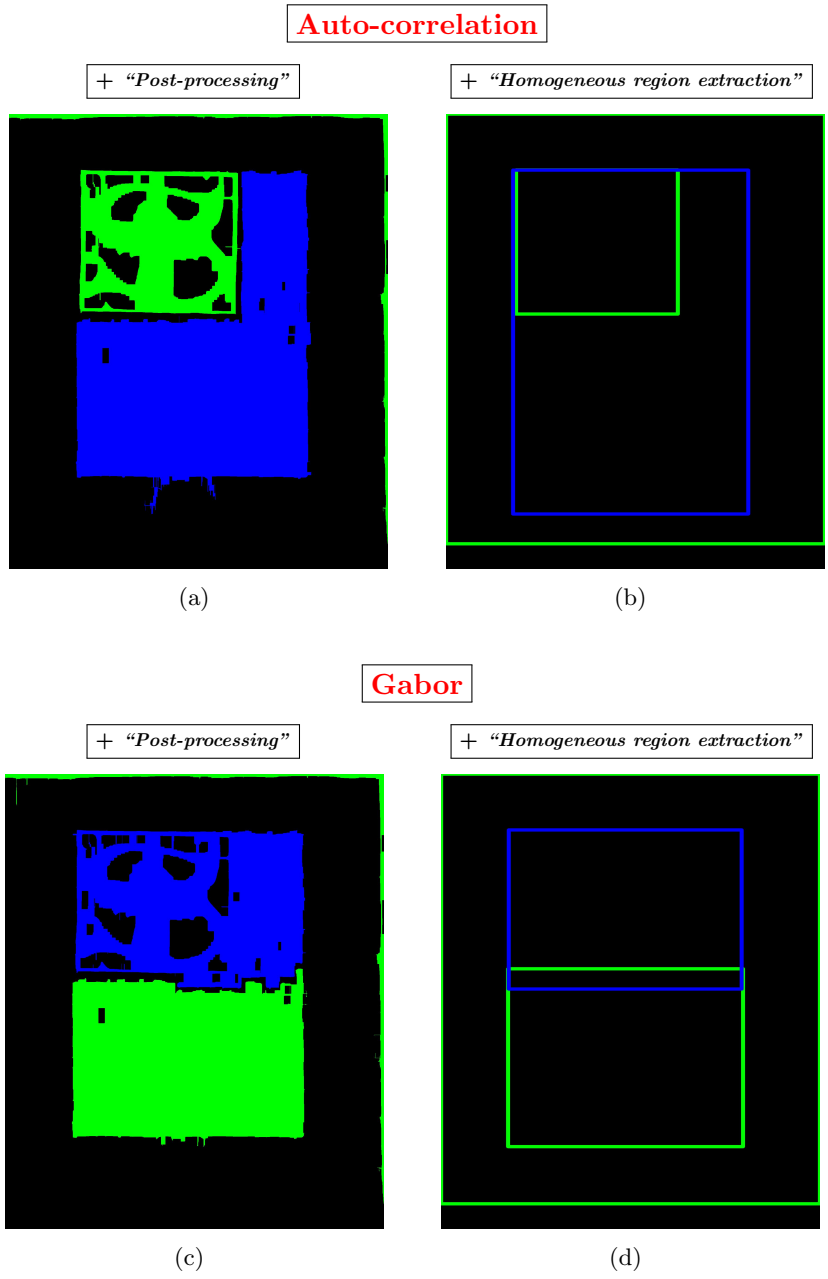


Figure 6.17.: Examples of visual results of the ***"Homogeneous region extraction"*** step, performed after the *"Post-processing"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"One font and graphics"*** HDI from the *"DIGIDOC-Texture dataset"*.

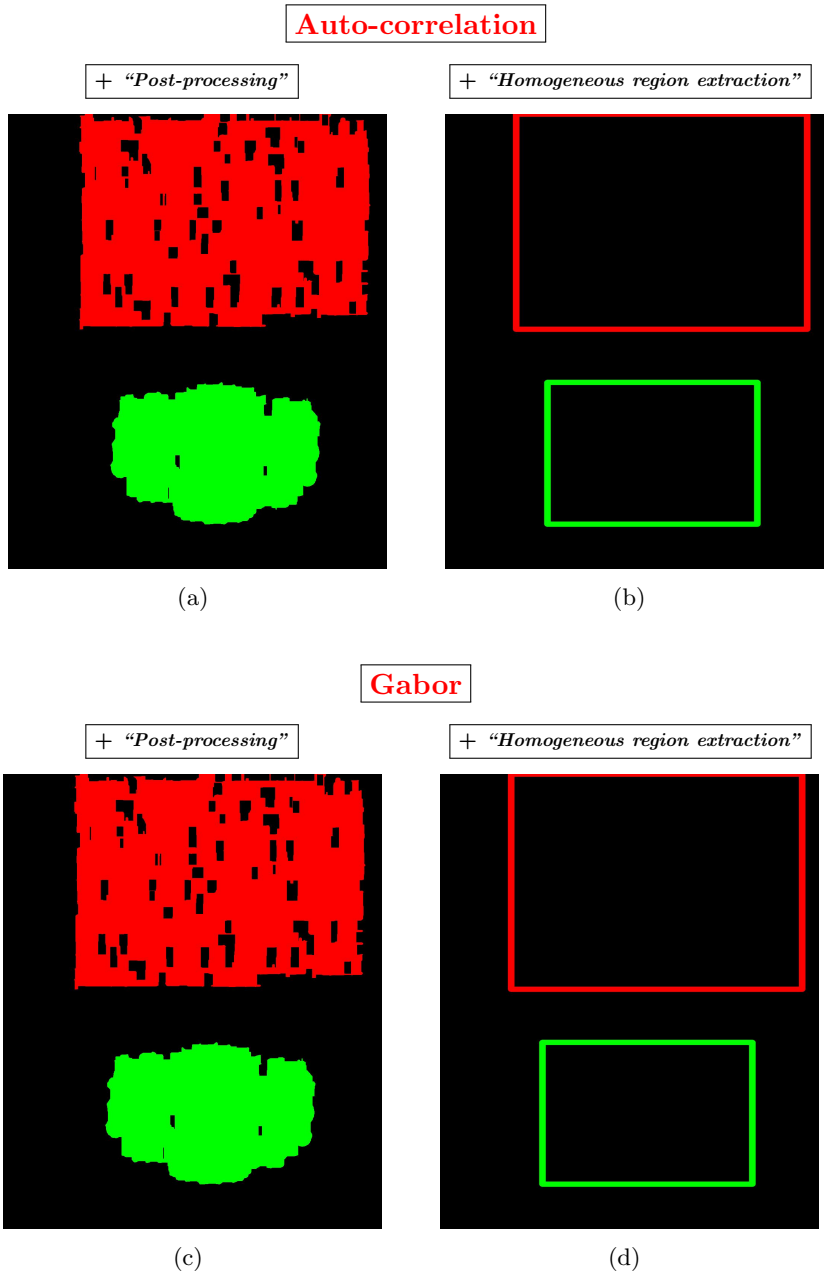
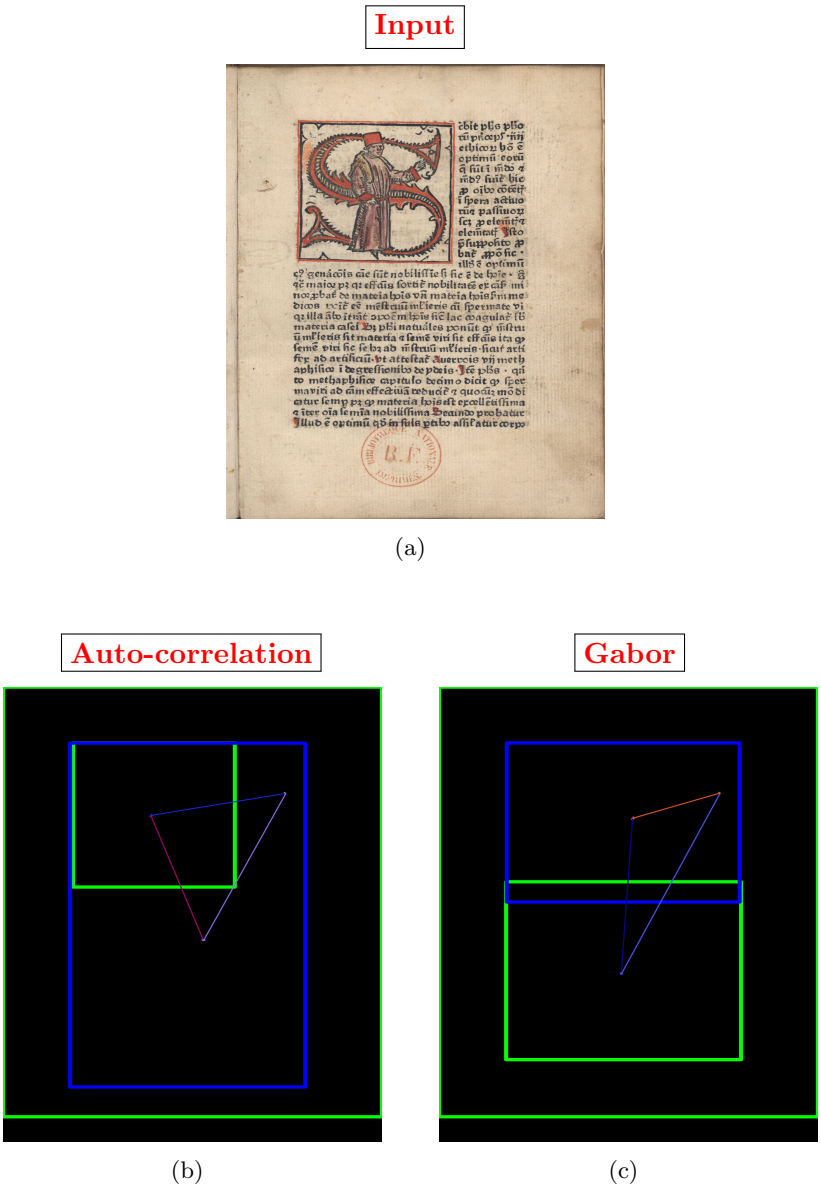


Figure 6.18.: Examples of visual results of the ***"Homogeneous region extraction"*** step, performed after the *"Post-processing"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics*"*** HDI from the *"DIGIDOC-Texture dataset"*.



(a)

(b)

(c)

Figure 6.19.: Examples of visual results of the **Structural signature generation** step, performed after the *"Homogeneous region extraction"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"One font and graphics"*** HDI from the *"DIGIDOC-Texture dataset"*.

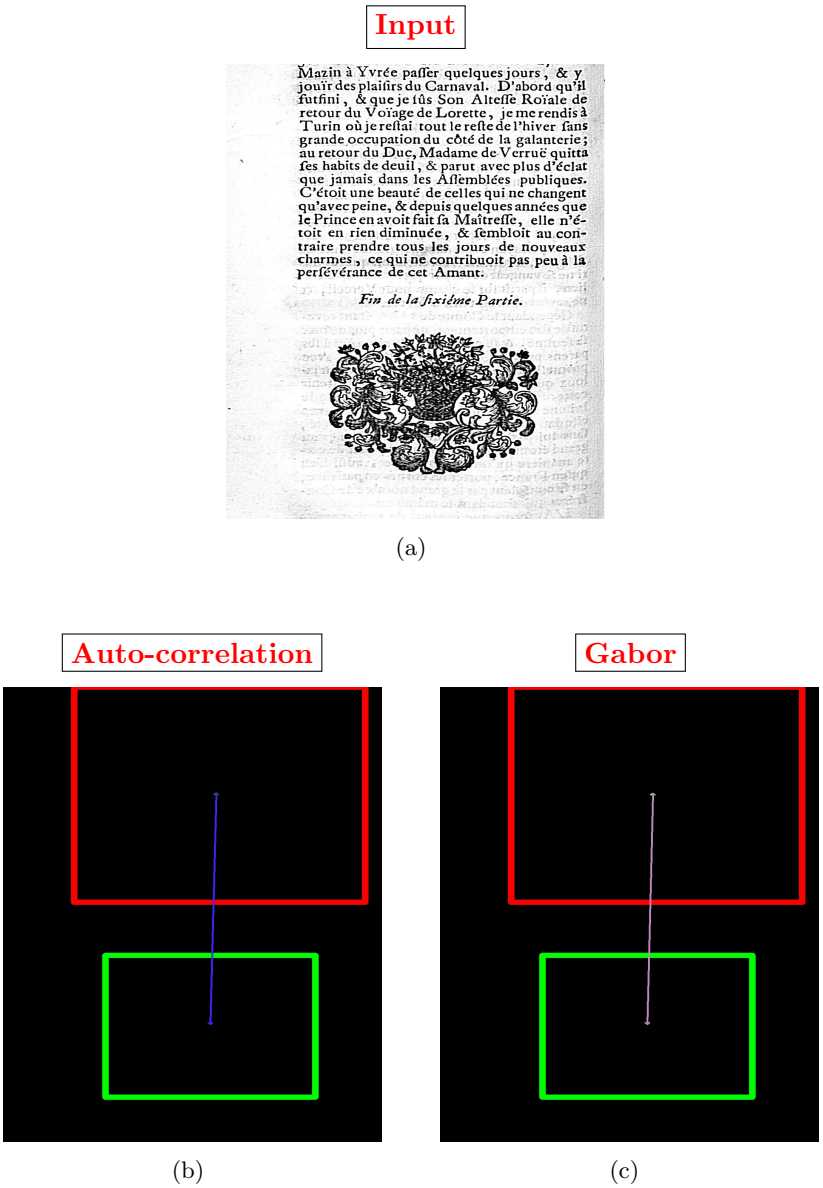


Figure 6.20.: Examples of visual results of the **Structural signature generation** step, performed after the *"Homogeneous region extraction"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics*"*** HDI from the *"DIGIDOC-Texture dataset"*.

Table 6.2.: Quantitative assessment of the ***"Pixel-labeling refinement"*** step using the results of the **auto-correlation** and **Gabor**-based pixel-labeling scheme with the ***"DIGIDOC-Texture dataset"*** by clustering and classification accuracy measures: Jaccard coefficient (J), purity per block metric (PPB), precision (P), recall (R), F-measure (F) and classification accuracy (CA). $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results. The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
Auto-correlation	One font and graphics	0.88	0.15	0.95	0.06	0.86	0.18	0.86	0.18	0.85	0.18	0.91	0.17
	Two fonts and graphics*	0.74	0.18	0.91	0.07	0.64	0.12	0.65	0.15	0.64	0.13	0.80	0.19
	Two fonts and graphics**	0.78	0.17	0.94	0.06	0.87	0.17	0.86	0.17	0.86	0.16	0.87	0.20
	Only two fonts	0.82	0.17	0.92	0.08	0.75	0.19	0.80	0.19	0.77	0.18	0.82	0.25
	Only three fonts	0.73	0.19	0.87	0.09	0.58	0.14	0.63	0.15	0.60	0.13	0.74	0.26
	Overall*	**0.79**	**0.17**	**0.91**	**0.08**	**0.71**	**0.16**	**0.73**	**0.17**	**0.71**	**0.15**	**0.82**	**0.22**
	Overall**	**0.80**	**0.17**	**0.92**	**0.07**	**0.76**	**0.17**	**0.78**	**0.17**	**0.77**	**0.16**	**0.84**	**0.22**
	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
Gabor	One font and graphics	0.88	0.18	0.96	0.06	0.90	0.16	0.86	0.19	0.88	0.17	0.88	0.23
	Two fonts and graphics*	0.70	0.16	0.93	0.06	0.70	0.16	0.66	0.13	0.67	0.14	0.75	0.17
	Two fonts and graphics**	0.81	0.16	0.98	0.04	0.91	0.13	0.88	0.16	0.89	0.14	0.89	0.21
	Only two fonts	0.82	0.22	0.94	0.09	0.89	0.15	0.81	0.22	0.84	0.19	0.83	0.24
	Only three fonts	0.60	0.19	0.88	0.09	0.67	0.17	0.62	0.18	0.64	0.17	0.68	0.19
	Overall*	**0.75**	**0.19**	**0.93**	**0.08**	**0.79**	**0.16**	**0.74**	**0.18**	**0.76**	**0.17**	**0.79**	**0.21**
	Overall**	**0.78**	**0.19**	**0.94**	**0.07**	**0.84**	**0.15**	**0.79**	**0.19**	**0.81**	**0.17**	**0.82**	**0.22**

Table 6.3.: **Difference values** in the computed clustering and classification accuracy measures when introducing the ***"Pixel-labeling refinement"*** step and without it into the **auto-correlation** and **Gabor**-based pixel-labeling scheme using the ***"DIGIDOC-Texture dataset"***: Jaccard coefficient (J), purity per block metric (PPB), precision (P), recall (R), F-measure (F) and classification accuracy (CA). The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	***J***	***PPB***	***P***	***R***	***F***	***CA***
Auto-correlation	One font and graphics	0.08800	0.04610	0.02770	0.0415	0.03380	0.06430
	Two fonts and graphics*	0.12990	0.07990	0.04560	0.05220	0.04840	0.09530
	Two fonts and graphics**	0.07690	0.04120	0.02990	0.03250	0.03150	0.03380
	Only two fonts	0.10290	0.07580	0.02830	0.07810	0.04940	0.04210
	Only three fonts	0.12440	0.09050	-0.04860	0.01560	-0.01800	0.11830
	Overall*	**0.11130**	**0.07308**	**0.01325**	**0.04685**	**0.02840**	**0.08000**
	Overall**	**0.09805**	**0.06340**	**0.00933**	**0.04193**	**0.02417**	**0.06463**
	Document content	***J***	***PPB***	***P***	***R***	***F***	***CA***
Gabor	One font and graphics	-0.00110	-0.00070	0.00020	-0.00210	-0.00100	0.01730
	Two fonts and graphics*	0.00080	-0.00030	-0.00040	-0.00020	-0.00040	0.00270
	Two fonts and graphics**	-0.00020	-0.00050	0.00020	-0.00020	0.00000	0.00000
	Only two fonts	0.00030	0.00010	0.00010	-0.00170	-0.00080	0.00380
	Only three fonts	0.00030	0.00010	-0.00010	-0.00230	-0.00110	0.00370
	Overall*	**0.00008**	**-0.00020**	**-0.00005**	**-0.00157**	**-0.00083**	**0.00687**
	Overall**	**-0.00018**	**-0.00025**	**0.00010**	**-0.00157**	**-0.00073**	**0.00620**

Table 6.4.: Quantitative assessment of the ***"Post-processing"*** step using the results of the *"Pixel-labeling refinement"* task performed on the **auto-correlation** and **Gabor**-based pixel-labeling scheme with the ***"DIGIDOC-Texture dataset"*** by clustering and classification accuracy measures: Jaccard coefficient (J), purity per block metric (PPB), precision (P), recall (R), F-measure (F) and classification accuracy (CA). $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results. The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
Auto-correlation	One font and graphics	0.90	0.16	0.99	0.02	0.78	0.24	0.75	0.28	0.76	0.26	0.92	0.17
	Two fonts and graphics*	0.87	0.17	0.98	0.04	0.66	0.18	0.65	0.20	0.65	0.19	0.88	0.19
	Two fonts and graphics**	0.83	0.18	0.99	0.03	0.85	0.21	0.83	0.24	0.84	0.22	0.91	0.17
	Only two fonts	0.87	0.18	0.96	0.08	0.75	0.23	0.73	0.26	0.74	0.24	0.89	0.20
	Only three fonts	0.78	0.20	0.92	0.10	0.56	0.18	0.54	0.23	0.54	0.21	0.78	0.26
	Overall*	**0.85**	**0.18**	**0.96**	**0.06**	**0.69**	**0.21**	**0.67**	**0.24**	**0.67**	**0.23**	**0.87**	**0.20**
	Overall**	**0.85**	**0.18**	**0.97**	**0.06**	**0.73**	**0.21**	**0.71**	**0.25**	**0.72**	**0.23**	**0.87**	**0.20**
	Document content	$\mu(J)$	$\sigma(J)$	$\mu(PPB)$	$\sigma(PPB)$	$\mu(P)$	$\sigma(P)$	$\mu(R)$	$\sigma(R)$	$\mu(F)$	$\sigma(F)$	$\mu(CA)$	$\sigma(CA)$
Gabor	One font and graphics	0.90	0.15	0.99	0.03	0.82	0.22	0.80	0.25	0.81	0.24	0.93	0.16
	Two fonts and graphics*	0.80	0.17	0.97	0.06	0.66	0.16	0.64	0.15	0.65	0.15	0.83	0.19
	Two fonts and graphics**	0.84	0.18	0.99	0.04	0.90	0.16	0.88	0.18	0.89	0.17	0.89	0.21
	Only two fonts	0.84	0.21	0.93	0.11	0.81	0.25	0.80	0.24	0.80	0.24	0.87	0.21
	Only three fonts	0.64	0.21	0.89	0.10	0.64	0.19	0.62	0.19	0.62	0.19	0.71	0.20
	Overall*	**0.80**	**0.19**	**0.94**	**0.07**	**0.73**	**0.21**	**0.71**	**0.21**	**0.72**	**0.20**	**0.83**	**0.19**
	Overall**	**0.81**	**0.19**	**0.95**	**0.07**	**0.79**	**0.21**	**0.77**	**0.22**	**0.78**	**0.21**	**0.85**	**0.19**

Table 6.5.: **Difference values** in the computed clustering and classification accuracy measures when introducing the ***"Post-processing"*** step and without it into the results of the *"Pixel-labeling refinement"* task into the **auto-correlation** and **Gabor**-based pixel-labeling scheme using the ***"DIGIDOC-Texture dataset"***: Jaccard coefficient (J), purity per block metric (PPB), precision (P), recall (R), F-measure (F) and classification accuracy (CA). The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	***J***	***PPB***	***P***	***R***	***F***	***CA***
Auto-correlation	One font and graphics	0.0189	0.0412	-0.0799	-0.1082	-0.0929	0.0012
	Two fonts and graphics*	0.1269	0.0744	0.0249	-0.0012	0.0126	0.0821
	Two fonts and graphics**	0.0589	0.042	-0.0115	-0.0272	-0.0187	0.0339
	Only two fonts	0.0492	0.0458	-0.0047	-0.064	-0.0296	0.0647
	Only three fonts	0.0515	0.0539	-0.0241	-0.0857	-0.0594	0.0445
	Overall*	**0.06163**	**0.05383**	**-0.02095**	**-0.06477**	**-0.04232**	**0.04813**
	Overall**	**0.04463**	**0.04573**	**-0.03005**	**-0.07128**	**-0.05015**	**0.03607**
	Document content	***J***	***PPB***	***P***	***R***	***F***	***CA***
Gabor	One font and graphics	0.0227	0.03	-0.0798	-0.0592	-0.0684	0.0423
	Two fonts and graphics*	0.1037	0.03	-0.0319	-0.016	-0.0242	0.0832
	Two fonts and graphics**	0.0315	0.0058	-0.0117	0.0032	-0.004	0.0047
	Only two fonts	0.0288	-0.0025	-0.077	-0.0116	-0.0408	0.0389
	Only three fonts	0.0365	0.0018	-0.0278	-0.0022	-0.0151	0.0262
	Overall*	**0.04792**	**0.01535**	**-0.05413**	**-0.02225**	**-0.03713**	**0.04765**
	Overall**	**0.02988**	**0.00820**	**-0.04908**	**-0.01745**	**-0.03208**	**0.02803**

Table 6.6.: Quantitative assessment of the ***"Homogeneous region extraction"*** step performed after the *"Post-processing"* task on on the **auto-correlation** and **Gabor**-based pixel-labeling scheme using the ***"DIGIDOC-Texture dataset"*** by computing three accuracy metrics: precision (P_{AR}), recall (R_{AR}) and Jaccard index (J). $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results. The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	$\mu(P_{AR})$	$\sigma(P_{AR})$	$\mu(R_{AR})$	$\sigma(R_{AR})$	$\mu(J_{AR})$	$\sigma(J_{AR})$
Auto-correlation	One font and graphics	0.96	0.12	0.90	0.20	0.88	0.23
	Two fonts and graphics*	0.92	0.14	0.78	0.21	0.76	0.23
	Two fonts and graphics**	0.91	0.15	0.75	0.23	0.72	0.25
	Only two fonts	0.97	0.09	0.80	0.25	0.79	0.27
	Only three fonts	0.91	0.17	0.73	0.29	0.69	0.31
	Overall*	**0.94**	**0.13**	**0.81**	**0.24**	**0.78**	**0.26**
	Overall**	**0.94**	**0.13**	**0.80**	**0.24**	**0.77**	**0.26**
	Document content	$\mu(P_{AR})$	$\sigma(P_{AR})$	$\mu(R_{AR})$	$\sigma(R_{AR})$	$\mu(J_{AR})$	$\sigma(J_{AR})$
Gabor	One font and graphics	0.97	0.11	0.90	0.19	0.88	0.22
	Two fonts and graphics*	0.91	0.16	0.78	0.20	0.75	0.22
	Two fonts and graphics**	0.91	0.15	0.77	0.20	0.74	0.22
	Only two fonts	0.97	0.09	0.79	0.25	0.78	0.26
	Only three fonts	0.92	0.13	0.73	0.25	0.68	0.28
	Overall*	**0.94**	**0.12**	**0.80**	**0.22**	**0.78**	**0.24**
	Overall**	**0.94**	**0.12**	**0.80**	**0.22**	**0.77**	**0.25**

Table 6.7.: **Difference values** in the computed accuracy metrics for the evaluation of the ***"Homogeneous region extraction"*** step performed after the *"Post-processing"* task between the **auto-correlation** and **Gabor**-based pixel-labeling scheme using the ***"DIGIDOC-Texture dataset"***: precision (P_{AR}), recall (R_{AR}) and Jaccard index (J). The *"Overall*"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics**"*. The *"Overall**"* value is obtained by averaging all the respective column values except the value of *"Two fonts and graphics*"*. *"Two fonts and graphics*"* represents the case when every font in the text has a different label in the ground-truth, and the clustering is performed by setting the number of types of content regions to 3 (graphics and two different text fonts). *"Two fonts and graphics**"* represents the case when all fonts in the text have the same label in the ground-truth, and the clustering is performed by setting the number of types of content regions equal to 2 (graphics and text).

	Document content	$\mu(P_{AR})$	$\mu(R_{AR})$	$\mu(J_{AR})$
Gabor-Auto-correlation	One font and graphics	0.00628	0.00008	0.00530
	Two fonts and graphics*	-0.00147	-0.00402	-0.00164
	Two fonts and graphics**	0.00357	0.01774	0.02021
	Only two fonts	-0.00057	-0.00935	-0.00063
	Only three fonts	0.01548	-0.00464	-0.00720
	Overall*	**0.00493**	**-0.00448**	**-0.00247**
	Overall**	**0.00619**	**0.00095**	**0.00299**

Chapter 7.

Application to DIGIDOC project: a structural signature for book page categorization

This chapter presents few applications of the proposed structural signature based on texture of digitized historical book pages in the context of DIGIDOC project. The proposed page signature is able to index, compare or categorize digitized historical book pages according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the historical document image content.

Contents

7.1. Introduction

Over the last few years, there has been tremendous growth in digitizing collections of cultural heritage documents. Thus, many challenges and open issues have been raised, such as information retrieval in digital libraries or analyzing page content of DHBs. The work presented in Chapter 6 proposes a structural signature based on texture, used for DHB page characterization. The proposed page signature integrates varying low-level features characterizing the different DI content components or blocks (*i.e.* text or graphic regions) on the one hand, and structural information describing the DI structure or layout on the other hand. This rich and holistic representation of the layout and content of the analyzed DHB page can be adapted to the user preferences and specified criteria through the extracted varying levels of information (e.g. by selecting only the information characterizing the HDI content and/or structure or by retrieving any useful information available for the subsequent use). The extracted varying low-level information corresponds to the extracted *(i)* texture features to characterize the DI typographical and graphical characteristics, *(ii)* shape, geometric and topological features to describe the shape and spatial relationships of the extracted components of DI contents and *(iii)* structural information to take into consideration the page layout or structure.

On the other side, our goal in the context of the DIGIDOC project is to develop tools for analyzing HDIs throughout the acquisition process, from scanning the document to knowledge representation and management of HDI content. Moreover, the ultimate goal of the DIGIDOC project is developing relevant ways of interacting with scanners by assisting the digitization operator to adjust automatically the best set of parameters (e.g. resolution, lightening, color calibration), detecting errors in the digitization process (e.g. blur, skewed or folded pages), providing appropriate assistance for document indexing (e.g. by recognizing automatically page types or breaks in a sequence of pages), *etc.* There is an absolute need to design "smart" digitizers which can limit manual intervention and perform easy and high quality digitization of DIs [9]. Therefore, to achieve better interaction with scanners, we need to design a computer-aided categorization tool, able to index or categorize DHB pages according to several criteria, mainly the layout structure, graphical properties or typographical characteristics of the HDI content. Thus, the key task in this work is to prove that it is possible to ensure automatic and relevant characterization and categorization of DHB pages without manual inspection or *a priori* knowledge regarding DI layout and content and with taking into consideration the particularities of HDIs.

As a matter of fact, the proposed page signature will be the data provided to a smart scanner to index or categorize DHB pages according to several criteria, mainly the layout structure, graphical properties or typographical characteristics of the HDI content. The DHB page categorization will be based on analyzing the different obtained signatures during the scanning process. To categorize and group DHB pages with similar layout and/or content, the obtained graph-based DHB page signature can be compared using a graph dissimilarity. Then, the evaluation of the proposed page signature has been carried out based on computing a distance matrix, whose elements represent the dissimilarity between the compared graphs. Indeed, the DHB pages can be compared by categorizing the designed signatures which model the layout and content of DHB pages. Figure 7.1 provides an overview of the context and an example of signature-based applications (*i.e.* finding pages in a DHB which contain similar content component or a group of patterns). In this work, we focus on investigating all the elements of the proposed graph-based signature to group DHB pages with similar layout and/or content. Nevertheless, it is worth noting that it is also possible to extend the scope of using the proposed graph-based signature to find pages in a DHB which contain similar content component or a group of patterns by means of sub-graph isomorphism paradigm.

Indeed, to deduce the similarities of DHB page structure or layout and/or content and subsequently to categorize and group DHB pages with similar layout and/or content, the proposed graph-based signature should be compared using a graph-matching paradigm and particularly the graph edit distance (GED) approach. The GED is used to measure the (dis)similarity between the proposed graph-based signatures and subsequently to group and categorize the pages that have

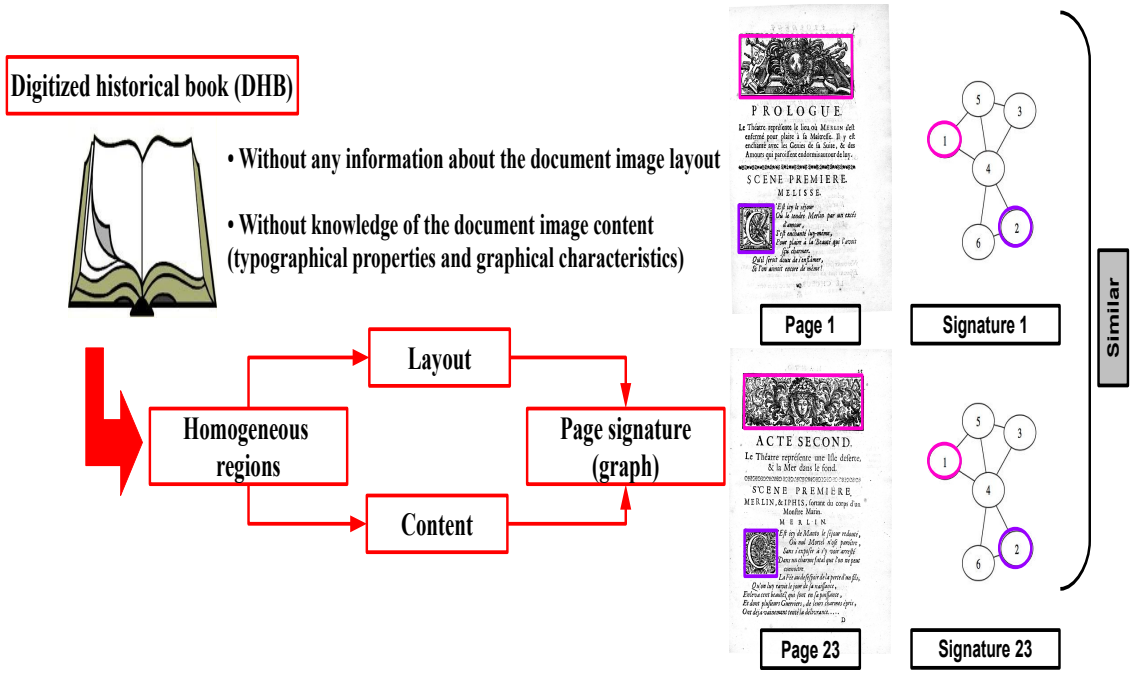


Figure 7.1.: Overview of the context and an example of signature-based applications (*i.e.* finding pages in a DHB which contain similar content component or a group of patterns).

similar content and/or structure [472].

Thus, based on the proposed structural signature, few applications for automatic categorization approach of DHB pages in the context of the DIGIDOC project are presented in this chapter. These applications help to manage effectively a corpus or collections of DHBs. As mentioned earlier (*cf.* Section 1.3), numerous applications based on the defined page signature can be proposed:

- Designing a smart or intelligent scanner by adapting or adjusting automatically the quality of the HDI scanning process with respect to the obtained page representations of DHB pages which can be classified according to several criteria (e.g. HDI content and/or structure, subsequent use). This would help ensuring an automatic adjustment of the digitization quality of historical collections with respect to the HDI content, layout and subsequent use,
- Modeling a computer-aided categorization tool, able to index, group or classify DHB pages according to several criteria, mainly the layout structure or typographic characteristics of the HDI content,
- Comparing the different DHBs according to several criteria, mainly the layout or content of their pages,
- Providing a DHB summary after determining the transition pages in a DHB which may correspond to the title pages of each chapter for example (*cf.* Figure 1.1),
- Retrieving pages in a DHB which have particular layout and/or content,
- Finding pages in a DHB that match specific criteria defined by a user,
- Detecting the scanning failure occurring during the digitization process (e.g. curvature, light),
- Collecting the empty or cover DHB pages, *etc.*

Among the numerous possible applications of the proposed DHB page signature mentioned above (e.g. structure the whole HDIs corpus, index, retrieve, compare or group HDIs), a thorough evaluation has been conducted in this work for assessing two possible signature-based applications:

1. Unsupervised DHB page classification to group or gather similar layout and/or content DHB pages,
2. DHB page stream segmentation to generate automatically a table of content/summary of the analyzed DHB.

Figure 7.2 illustrates the two analyzed and evaluated categorization applications of the proposed DHB page signature, unsupervised page classification and page stream segmentation.

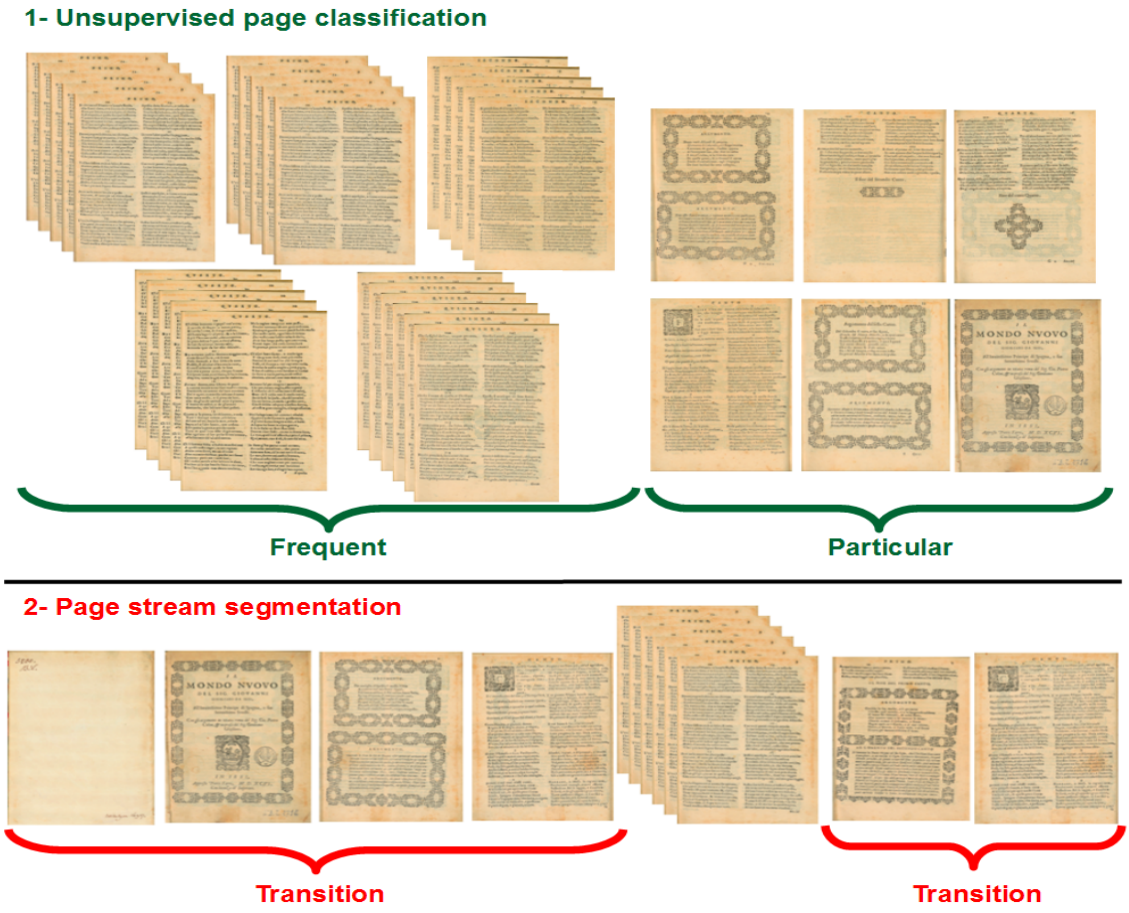


Figure 7.2.: Illustration of the two analyzed and evaluated categorization applications of the proposed DHB page signature, unsupervised page classification and page stream segmentation.

As a consequence, in this chapter we detail these two applications of the proposed page signature along with the experiments and evaluations necessary to assess their performance. This evaluation has been carried out based on:

- Computation of GEDs between the different graph-based DHB page signatures, that can be used to retrieve similar pages in a HDI database query tool.
- DHB page categorization by analyzing the computed GEDs between the different graph-based DHB page signatures.

The assessment of the other applications of the proposed DHB page signature, cited earlier, will be among our future prospects.

The two analyzed and evaluated applications are independent of the layout and content of the analyzed DHB pages, and hence, they are applicable to a large variety of DHBs. Indeed, our proposed approach does not assume *a priori* knowledge regarding HDI content and structure. In order to test the performance of the proposed signature, a detailed experimental evaluation on a large variety of HDIs has been carried out in two different signature-based applications.

Therefore, based on the work presented in the previous chapters (*cf.* Chpaters 4, 5 and 6), a texture-based structural signature for characterization and categorization of DHB pages is illustrated in Figure 7.3.

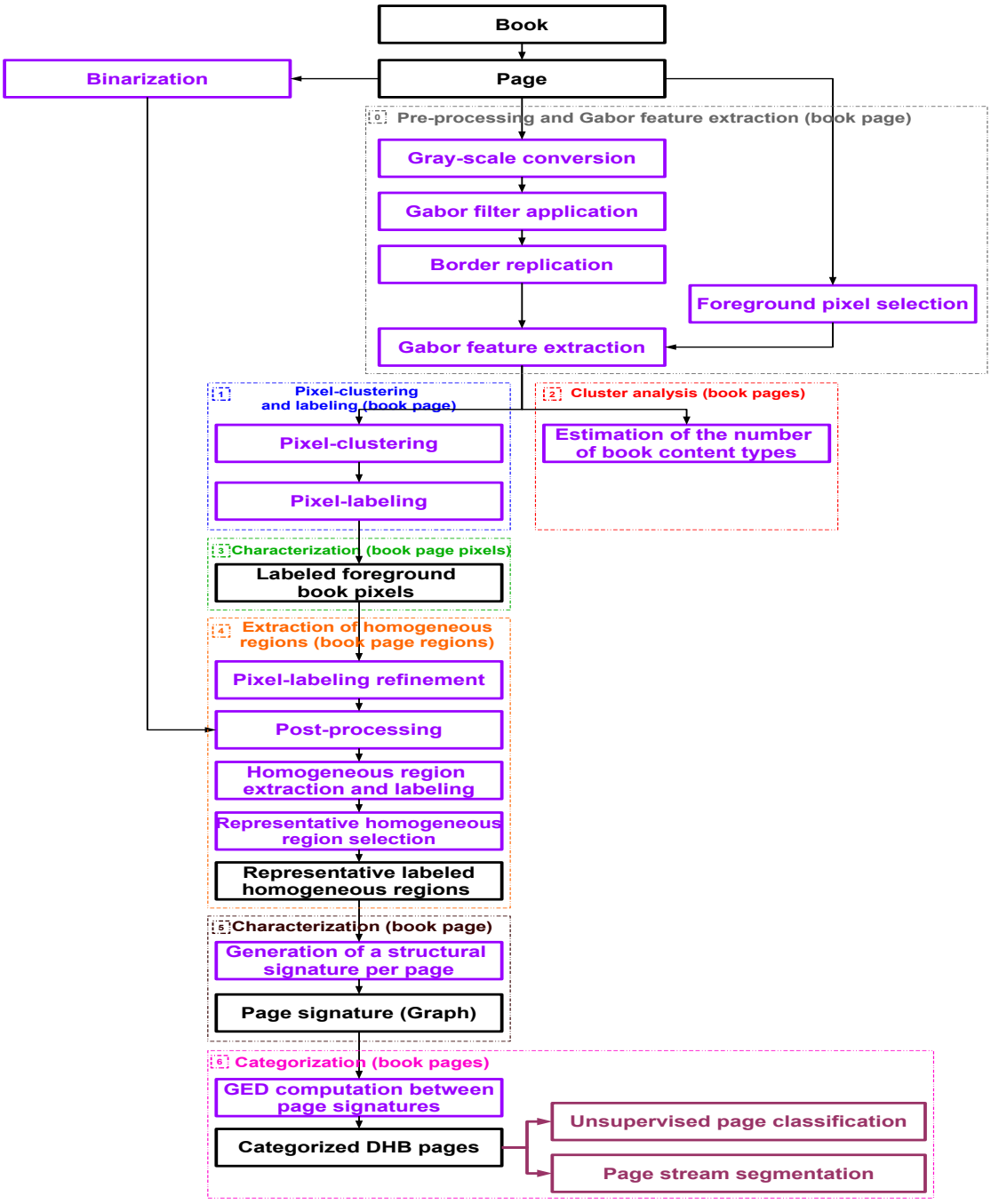


Figure 7.3.: Detailed schematic block diagram of the proposed texture-based structural signature for characterization and categorization of DHB pages.

First, Chapter 4 states that Gabor-based approach performs considerably better in segmenting

HDIs. Then, Chapter 5 presents a framework to investigate the use of texture as a tool for determining automatically the number of book content types in a DHB and segmenting its contents by extracting and analyzing texture features independently of the layout of the pages. Finally, Chapter 6 proposes a structural signature based on Gabor features, used for DHB page characterization. The proposed signature is based on varying low-level features (*i.e.* Gabor, shape, geometric and topological descriptors) and a structural signature. Therefore, by integrating the Gabor-based signature presented in Chapter 6 after the proposed Gabor-based pixel-labeling framework presented in Chapter 5, a structural signature based on Gabor features of DHB page is analyzed and evaluated in this chapter. This signature is defined according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the content of the DBH into consideration. This chapter illustrates the potential of the proposed signature by evaluating two possible signature-based applications, unsupervised page classification and page stream segmentation for DHB page categorization.

Therefore, the work presented in this dissertation consists of a complete system which can be divided into two parts:

1. ***DHB page characterization***:
   The first part of this work is used to generate the graph-based signature for DHB page characterization, is composed of the following seven tasks:
   a) Pre-processing and Gabor feature extraction (*Step* 1, *cf.* Sections 4.4.1.1 and 4.4.1.2),
   b) Estimation of the number of DHB content types (*Step* 2, *cf.* Section 5.3.1.2),
   c) Pixel-clustering and labeling (*Step* 3, *cf.* Section 5.3.2),
   d) Pixel-labeling refinement (*Step* 4, *cf.* Section 6.3.1),
   e) Post-processing (*Step* 5, *cf.* Section 6.3.2),
   f) Extraction of representative homogeneous regions (*Step* 6, *cf.* Section 6.3.3),
   g) Generation of a structural signature per page (*Step* 7, *cf.* Section 6.3.4).

2. ***DHB page categorization***:
   Since the characterization of the DHB page layout and content is performed using the proposed graph-based signature, the categorization task of the DHB pages can be carried out by comparing the different graph-based signatures. A thorough evaluation has been conducted in this work for assessing two signature-based applications:
   a) Unsupervised DHB page classification (*cf.* Section 7.4.1),
   b) DHB page stream segmentation (*cf.* Section 7.5).

Figure 7.4 illustrates the detailed schematic block diagram of the proposed approach used to generate the graph-based signature for DHB page characterization.

The remainder of this chapter is organized as follows: Section 7.2 reviews the different techniques and algorithms of graph-matching paradigms. In Section 7.3, the used GED computation by means of a binary linear programming (BLP) is described briefly. Section 7.4 presents two applications of the proposed DHB page signature for DHB page categorization, unsupervised DHB page classification (*cf.* Section 7.4.1) and DHB page stream segmentation (*cf.* Section 7.4.2). In Section 7.5.1, we discuss the obtained performance of each application of the proposed structural signature for DHB page categorization by computing several accuracy metrics. Moreover, to evaluate the performance of the proposed signature-based approach for DHB page characterization on a DHB of 322 ground truthed one-page HDIs, qualitative and quantitative evaluation of the different steps of its extraction are presented. In addition, qualitative results using the designed GUI tool for DHB page categorization are also given to demonstrate the performance of the proposed DHB page categorization. Our discussion and conclusions are presented in Sections 7.6 and 7.7, respectively.

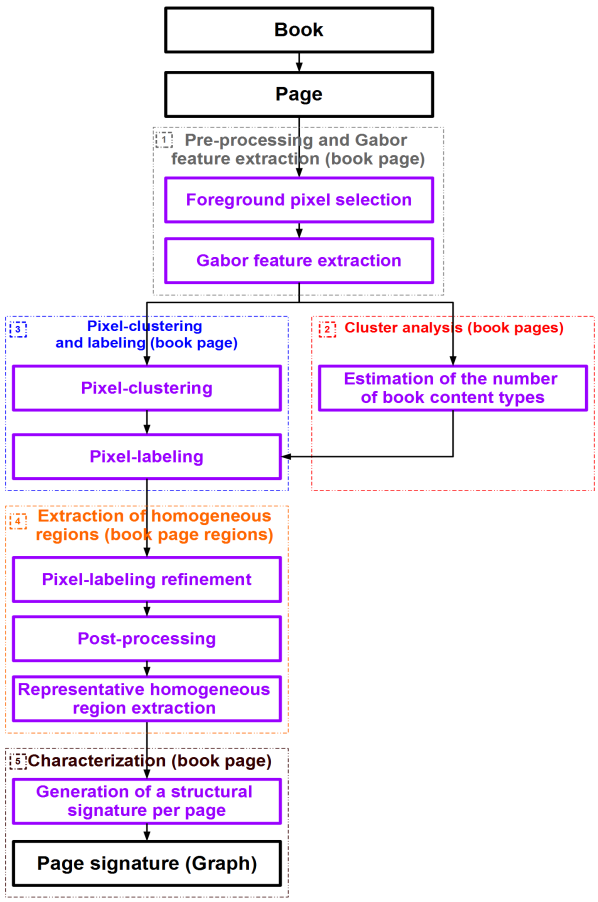


Figure 7.4.: Detailed schematic block diagram of the proposed approach used to generate the graph-based signature for DHB page characterization.

7.2. Related works

Since the goal of this chapter is to illustrate the effectiveness of the proposed graph-based page signature by assessing possible signature-based applications of DHB page categorization, graph-matching paradigm is obviously required to compare the involved signatures. As a matter of fact, this section reviews the different techniques and methods used to solve graph-matching paradigms, with a particular focus on those related to GED computation.

7.2.1. Graph-matching paradigm

In the last decades, graph usage has grown significantly due to the inherent flexibility and generality of graph structures and the numerous advantages over other topological representation formalisms in pattern recognition fields previously detailed in Chapter 6 (*cf.* Section 6.2.3) comparing other spatial representation formalisms. Indeed, due to the fact that graphs ensures the modeling of different types of data, graph-based applications on several pattern recognition fields have been developed. Critical to this intensive emergence of graph use are the huge volume of graph data which have become available and the open issues related to the development of effective and efficient methods to perform graph mining [473, 474, 475], graph clustering [476, 477] and graph classification [478, 479, 480].

Nevertheless, it is well-known that computing the dis(similarity) measure among graphs is considered as the crucial crossroad of these different graph-based applications and particularly for machine learning issues. Usually, the graph-matching issue is addressed by computing the dis(similarity)

measure between pairs of graphs. The graph-matching algorithms are based on computing the distance between two graphs. The lower the distance values, the more the graphs can be considered similar.

Silva *et al.* [481] confirmed that graph-matching is still a highly complex open issue related to the use of the exact or inexact "approximate" graph-matching approaches. He stated that the complexity of the exact graph-matching approaches has not been solved yet, contrary to the sub-graph-matching or approximate graph-matching which are NP-complete problems [482, 483]. As a consequence, the graph-matching issue is tackled in different ways by many researchers. Unfortunately, there is no generic solution capable of addressing efficiently this issue. A large number of algorithms have been proposed in the literature in this respect. The state-of-the-art methods addressing the computation of the dis(similarity) measure among graphs can be categorized into two classes:

1. ***Embedding-based methods***
   The embedding-based methods are processed by projecting the input graphs into real vector space. The idea of these methods consists in embedding a part of a graph into a vector feature space, for example a numerical vector. These methods have the advantage of benefiting from the access to the rich repository of algorithmic tools for pattern recognition (*i.e.* the distance computation used for vector representations). Moreover, they ensures the reduction of the computation of a distance between two graphs to a distance between two vector (*i.e.* linear complexity). Thus, they are computationally effective, since they do not require a complete matching process. The ultimate objective of these methods is to integrate the graph structure into a computationally efficient and mathematically convenient feature vector which is not a straightforward task. As a matter of fact, they might affect the effectiveness of the graph-matching process. On the other hand, the issue of determining the adequate vector representations for graphs is absolutely a non-trivial task due to the representational power of graphs which is clearly higher than that of feature vectors (*i.e.* a supplementary off-line time is required for database indexing) [439]. Indeed, a loss of information can be induced due to the mis-representation of the relational properties in vector space, when the projection of the input graphs into vector space is performed using a vector representation (*i.e.* non-bijectivity between the graph and vector spaces). The embedding-based methods can also be classified into two categories:

   a) ***Based on implicit projection***
      Among the embedding-based methods based on implicit projection, we mention as examples methods based on using graph kernels [484, 485, 486]. The graph kernel methods are used for classification, transformation and clustering of vector space embedded graphs, *etc.* A number of graph kernels have been designed which can be classified into four kinds: diffusion, convolution, walk and other additional kernel methods [487, 488, 489, 490, 439].

      i. ***Convolution kernels***
         The convolution kernels are processed by inferring the similarity of complex patterns from the similarity of their components. The ANOVA kernel [491] or graphlet kernel [492] are two standard statistical convolution kernels.

      ii. ***Diffusion kernels***
         The Diffusion kernels are processed by defining a base similarity measure which is used to construct a valid kernel matrix. This base similarity measure is needed to repeatedly fulfill the condition of symmetry. It can be defined for any kind of objects [493].

      iii. ***Walk Kernel methods***
         The walk Kernel methods are based on the analysis of random walks in graphs by measuring the similarity of two graphs. The similarity of two graphs is defined by

the number of random walks in both graphs that have all or some labels in common [444].

iv. ***Other additional kernels***
These other additional kernels are based on identifying identical sub-structures in two graphs, such as common sub-graphs, sub-trees and cycles [494, 495].

The high representational power of graphs and large repository of algorithmic tools available for feature vector representations of objects are considered two main advantages of using the graph kernel methods. Nevertheless, these kernels can only be used in specific applications and kinds of graphs.

b) ***Based on explicit projection***
The embedding-based methods based on explicit projection are based on computing a feature vector for each graph. The vector features can be deduced from the appearance frequencies of a specific sub-structure [496, 497, 498] or from a spectral analysis of graphs [499, 448, 500].

i. ***Methods based on the appearance frequencies of a specific sub-structure***
The embedding methods based on the appearance frequencies of a specific sub-structure focus mainly on the sub-graph extraction task. For instance, Barbu *et al.* [497] proposed for the clustering of DI. The DI representation is deduced by counting the occurrences of structural patterns. The pattern lexicon is constituted of the frequent sub-graphs in the structural representations of DIs. One of the major drawbacks of this approach is that as the constituted pattern lexicon is specific for a DI dataset, it must be regenerated in order to represent DIs from another dataset. Sidère *et al.* [498] proposed a vector representation of graphs based on pattern frequency, by integrating labeling information, to classify symbols and letters. Nevertheless, these methods require an off-line time for database indexing. In addition, the performance of the proposed approaches depends on the choice of parameters, such as the size of lexicon which must be specified in advance.

ii. ***Spectral methods***
The spectral methods are based on the following observation, first, the eigen-values and eigenvectors of the adjacency or Laplacian matrix of a graph are known to be invariant to vertex permutations. Hence, if two graphs are isomorphic (*cf.* Section B.8), their structural matrices will have the same eigen-decomposition. As a consequence, by means of the eigen-decomposition, the underlying graphs can be represented and compared with some features derived from their eigen-decomposition. The main disadvantage of the spectral methods is, that they are sensitive to few structural errors, such as missing or spurious vertices [501].

2. ***Matching-based methods***
The matching-based methods are based on determining the similarity between two graphs by computing and quantifying the "best" matching between them. There are several types of matching algorithms used to compute and quantify the "best" matching between them. Recently, numerous research studies have focused on proposing efficient and effective graph-matching algorithms [441, 502]. A huge number of algorithms have been proposed in the literature to reduce the computation and complexity requirements to search the most similar graph or sub-graph [440, 441]. These algorithms can be categorized according to the kinds of constraints that must be respected or relaxed (e.g. determining the maximum common sub-graph and/or minimum common sub-graph has been used to deduce a graph distance metric [503, 504, 505]). On the other side, the graph-matching algorithms can be classified into two categories, based on exact isomorphism and error-tolerant graph-matching algorithms (*cf.* Section B.8). Indeed, in the real world issues the involved data for graph-matching

paradigm are potentially biased information which can affect the labels and/or topology of the structural representations under consideration. As a consequence, the error-tolerant graph-matching issues have gained a great attention of many researchers in the pattern recognition and analysis fields. A graph-matching algorithm is called error-tolerant one if few matching approximations can be made on the topology and/or attributes. The adjacency matrix eigen-decomposition [506, 507] and graduated assignment methods [508, 509] are two examples of error-tolerant graph-matching algorithms. The major drawback of these algorithms is that many critical conditions must be satisfied and numerous heuristics should be established (e.g. *(i)* The pair of graphs to be matched must be nearly isomorphic; *(ii)* The eigen-values of the adjacency matrix of each graph have to be single and isolated enough to each other). Another well-known example of error-tolerant graph-matching algorithms is the GED [472].

7.2.2. Graph edit distance

The GED which corresponds to the minimum cost associated to an error correcting graph-matching, has been intensively investigated since it is on the crossroad of different pattern recognition and computer vision fields. It is used to measure the (dis)similarity between graphs [13]. Hence, in this work the GED approach is well suited to analyze and evaluate the different signature-based applications for DHB page categorization (*i.e.* the obtained graph-based DHB page signatures can be compared using a graph dissimilarity by means of GED).

The GED deals with the computation of the minimum-cost sequence of the basic graph editing operations (e.g. substitution, deletion and insertion of vertices or edges) to transform a graph to another one. The GED has to be set up based on the costs of the elementary edit operations (substitution, deletion and insertion). These costs are functions of the label of vertices/edges. In Appendix B and particularly in Section B.8, a detailed description of the GED approach has been carried. The major advantage of the GED is its generality to be arbitrarily applied to attributed graphs and to any type of graphs, including hyper-graphs. Moreover, there are neither critical conditions/restrictions to be satisfied nor heuristic information to be established [472]. GED has be mainly used to address various issues related to graph classification [478, 479, 480]. However, the main disadvantage of GED is its computational complexity which is exponential in the number of vertices of the involved graphs. As a consequence, GED is only effective and efficient for graphs of small size. Nevertheless, to tackle this issue, several fast suboptimal algorithms have been proposed to tackle the efficiency limitation of GED by proposing solutions which are designed to enable quick calculation of GEDs [510, 511, 512, 513, 514, 515, 516]. Other optimal methods have also been proposed for the efficient computation of GED [517, 518]. Zeng *et al.* [519] stated that the GED computation is a NP-hard approach. Indeed, using exact approaches when computing GED for large graphs is prohibitively difficult (*i.e.* exact approaches to compute GED are effective for small graphs). Hence, approximate approaches by means of upper and lower bounds of the exact GED computation have become the best alternative when computing exact GED for large graphs. Numerous surveys of GED approaches have been proposed in the literature [441, 520, 521]. Consequently, GED approaches can be classified into two categories, exact and approximate/inexact approaches.

7.2.2.1. Exact approaches

The first category of the GED approaches is based on exact computation of GED. A^* is certainly the most well-known algorithm, used to compute exactly GED [522]. It is based on the exploration of the tree of solutions. Each node in this tree represents the partial edition of the involved graph. On the other side, a tree leaf represents an edit path which transforms one of the input graph into the other one. The exploration of the tree is based on developing relevant methods for the estimation of GED. Indeed, the estimation of GED consists in determining the sum of the cost associated to the partial edit cost and the cost of the remaining path which is given by a heuristic. An

optimal path from the root node to the leaf one is identified based on the following statement: the estimation of the future cost is lower than or equal to the real cost. Nevertheless, if the estimation of the future cost is set to zero the whole tree of solutions will be explored (*i.e.* the smaller the difference between the estimation and real future cost, the fewer nodes will be explored). On the other side, by computing the real cost for the remaining edit path, an exponential time is required. The A^*-based methods proposed in the literature differ depending on the defined heuristics for the future cost estimation [517, 523]. These heuristics have been established to find an optimal trade-off between the approximation quality and computation time. Another well-known algorithm based on a binary linear programming (BLP) approach was proposed to compute exactly GED [524, 480]. For instance, Justice and Hero [480] proposed a BLP formulation of the GED for unweighted, undirected graphs. Their method aims to determine the permutation matrix which minimizes the cost of transforming a graph to another one.

7.2.2.2. Approximate approaches

Since the exact computation of GED is only effective and efficient for graphs of small size. Indeed, the computational complexity of the exact computation of GED is exponential in the number of vertices of the involved graphs. Meanwhile, considerable efforts have been undertaken to propose numerous computations of approximations in polynomial time. As an example, Justice and Hero [480] proposed a lower and upper bounds of the exact GED which can be computed in $\mathcal{O}(n^7)$ and $\mathcal{O}(n^3)$, respectively. Riesen and Bunke [513] used a cost matrix for vertex substitution, insertion or deletion to determine the vertex assignment by applying the Munkres' algorithm [525]. The vertex assignment ensures the inference of an edit path which transforms one graph into another one and whose associated cost is an upper bound of the exact GED. Their method has a complexity of $\mathcal{O}((n_1 + n_2)^3)$ in n_1 and n_2 which denote the number of vertices of the two involved graphs.

Other approximate approaches are derived from the exact ones. For instance, Neuhaus *et al.* [514] proposed two simple and effective approximations of a standard GED by using A^*-based algorithm to sub-optimally compute GED in a faster way. The first approximation which is called A^*-Beamsearch, is based on pruning the tree of solutions by limiting the number of concurrent partial solutions to the p most promising ones. The parameter p which defines the number of concurrent partial solutions to keep, is determined by finding an optimal trade-off between the approximation quality and combinatorial cost. The first approximation provides a valid edit path and its associated cost by setting an upper bound of the exact GED. Nevertheless, the identified edit path can be not the optimal one (*i.e.* an optimal edit path may be filtered off and removed in the earlier steps of the algorithm). The second approximation which is called A^*-Pathlength, is based on providing a higher exploration priority to long partial edit paths in order to have a prompt access to a leaf vertex. Riesen *et al.* [526] improved their proposed approximate GED, previously published in [513], by means of genetic algorithms and vertex assignments for search procedure. The vertex assignment are computed using bipartite graph-matching approach as an initialization step for a genetic algorithm. The bipartite graph-matching approach derives an edit path from any vertex assignment and subsequently computes its cost [513]. Another recent approximate GED approach based on Hausdorff matching was proposed by Fischer *et al.* [523]. It integrates in the A^*-based algorithm a heuristic based on a modified Hausdorff distance. The modified Hausdorff distance has a time complexity of $\mathcal{O}(n_1 \times n_2)$ in n_1 and n_2 which denote the number of vertices of the two involved graphs.

Other kind of approximate GED approaches has been proposed based on probabilistic framework [515, 527]. For instance, Myers *et al.* [515] proposed a framework for comparing and matching corrupted relational graphs by means of Bayesian GED. They modeled the probability distribution for structural errors in vertex assignments. The objective is to find the vertex assignment that maximizes the *a posteriori* probability considering vertex attributes. Nevertheless, this kind of approximate GED approaches can neither define bounded heuristics, nor exploit the use of algorithms to prune the tree of solutions by efficiently prioritize its exploration in the A^*-based algorithm.

7.3. Graph edit distance using an optimized binary linear programming

It is important to underline that the proposed characterization approach of DHB pages by means of a graph-based signature (including the definition of the node and edge labels of the obtained graph-based signature after extracting the representative homogeneous regions), forms an integral part of our concrete contribution. On the other side, an approximate GED approach has been performed using an optimized formulation of binary linear programming (BLP) by means of the GEM++[1] tool for comparing the different DHB page signatures. The GEM++ tool was proposed by the LITIS and LI laboratories for solving graph-matching paradigm using an optimized formulation of BLP by means of a lower bound of the exact GED to model approximate GED paradigm [528].

A BLP is a derivative of integer linear programming (ILP) where the variables are binary. Indeed, a set of binary variables is used to define an edit path between the graphs G^1 and G^2 formulated in a BLP problem to find an edit path on graph G^1 to make it isomorphic to the graph G^2 by means of GED. In Appendix B and particularly in Section B.8, a brief review of the basic definitions and concepts related to graphs and a detailed description of a standard GED approach have been carried. To compute the GED between the graphs G^1 and G^2, an edit path is defined by a set of binary variables on graph G^1 to make it isomorphic to the graph G^2. Three types of elementary edit operation are defined when using GED approach to match the two graphs G^1 and G^2:

1. The ***substitution*** of the label of a vertex (resp. an edge) of G^1 with the label of a vertex (resp. an edge) of G^2,
2. The ***deletion*** of a vertex (resp. an edge) from G^1,
3. The ***insertion*** of a vertex (resp. an edge) of G^2 in G^1.

To find the best admissible edit path between the graphs G^1 and G^2, an overall cost can be deduced. This overall cost must be minimized by means of an objective function defined by a set of binary variables on graph G^1 to make it isomorphic to the graph G^2. Afterwards, our goal is to find the optimal solution that minimizes the objective function and respects several constraints. This can be expressed in terms of a BLP problem which is used to model approximate GED paradigm.

Few domain and linear inequality constraints must be respected to have admissible edit path solutions of the defined BLP that minimizes the objective function (*cf.* equation B.88) applied on the graph G^1 to make it isomorphic to the graph G^2. A solution is considered as admissible if and only if the defined domain constraints (*cf.* equations (B.89) and (B.90)) and linear inequality constraints (*cf.* equations (B.82), (B.83), (B.84), (B.85), (B.86) and (B.87) in Table B.5), related to the involved edit path solution are respected (*cf.* Appendix B and particularly Section B.9.2). An admissible edit path solution of the optimized BLP that minimizes the objective function applied on the graph G^1 to make it isomorphic to the graph G^2, is given based on the optimized BLP formulation of GED which is illustrated in Table B.6.

In this work, an approximate GED approach is used to provide sub-optimal solutions with unbounded errors. Indeed, a lower bound solution of the minimization problem is obtained by using a continuous relaxation of the optimized BLP formulation (*i.e.* the constraints remain unchanged while the variables used when setting the domain constraints are defined in the continuous space $[0, 1]$). The continuous relaxation is considered as an heuristic used to compute an approximation of the optimal objective value in conjunction with a branch-and-cut algorithm when exploring the tree of solutions. This ensure the reduction of the number of explored solutions in the tree of solutions and subsequently the computational complexity.

Finally, to solve the optimized BLP formulation of GED (*cf.* Table B.6) and determine an admissible edit path solution, a dedicated mathematical solver which is called Gurobi[2], is used based on a branch-and-cut algorithm in conjunction with a continuous relaxation. Indeed, the used

[1] `http://litis-ilpiso.univ-rouen.fr/ILPIso/gem++.html`
[2] `http://www.gurobi.com/`

solver finds the best admissible edit path solution in terms of the optimized BLP that minimizes the objective function applied on the graph G^1 to make it isomorphic to the graph G^2. The used approximate GED in this work has a polynomial time complexity. Nevertheless, the size of the graphs under consideration remains a significant problem for time complexity. Therefore, a limited number of vertices in the involved graphs is still necessary to have reduced computational complexity of the proposed GED approach.

In Appendix B and particularly in Section B.9, a detailed description of how an overall cost can be deduced by using a BLP and applying an edit path on graph G^1 to make it isomorphic to the graph G^2.

7.4. Categorization of digitized historical book pages

To categorize and group DHB pages with similar layout and/or content, the obtained graph-based DHB page signature can be compared using a graph dissimilarity. In our experiments, we use a graph dissimilarity tool (*cf.* Section 7.3) which provides an approximation of a standard GED. The assessment of other potential graph dissimilarity approaches, cited earlier, will be among our future prospects, since in this work the computational complexity of the approximate GED is reduced. Indeed, we have a limited number of vertices in the obtained graphs (*i.e.* up to 11 vertices).

The approximate GED is used to measure the (dis)similarity between the obtained graph-based DHB page signatures [13]. The approximate GED deals with the computation of the minimum-cost sequence of the basic graph editing operations (e.g. substitution, deletion and insertion of vertices or edges) to transform a graph to another one. The approximate GED has to be set up based on the costs of the elementary edit operations (substitution, deletion and insertion). These costs are functions of the label of vertices/edges. The weight of each feature composing the label has been set after a statistical analysis of the feature variations in order to give the same importance to texture features and shape/geometric/topological descriptors (*cf.* Table 7.1). Same weight of each feature composing the label are assigned to the deletion and insertion graph editing operations, since these two basic graph editing operations are reversible.

Table 7.1.: Assigned weights to the basic graph editing operations (substitution, deletion and insertion) for the vertex and edge attributes of the proposed structural signature.

			Weight	
	Id.	**Attribute**	**Substitution**	**Deletion/insertion**
Vertex	$A^v_{1\to 46}$	Topological, geometric and shape attributes	$\frac{1}{46\ \sigma_{A^v_{1\to 46}}}$	$\frac{1}{46\ \sigma_{A^v_{1\to 46}}}$
	$A^v_{47\to 238}$	Texture attributes	$\frac{1}{192\ \sigma_{A^v_{47\to 238}}}$	0
Edge	A^e_1	Absolute difference between the two extracted region centroids in the x-axis	1	1
	A^e_2	Absolute difference between the two extracted region centroids in the y-axis	1	1
	A^e_3	Edge force	1	0

Therefore, the evaluation of the proposed page signature has been carried out based on firstly computing a distance matrix, whose elements represent the dissimilarity between the compared graphs. The dissimilarity corresponds to the GED, normalized with respect to the graph size. Indeed, for a fixed number of edit operations needed to transform one graph into another, the dissimilarity is higher if the graphs are small, and lower if the involved edit operations only affect

a tiny portion of a large graph. Then, by analyzing the elements of the resulting distance matrix (M^g), two following applications are targeted, unsupervised page classification and page stream segmentation.

7.4.1. Unsupervised page classification

Firstly, an unsupervised classification task using the HAC algorithm, is performed on all elements of M^g ($m^g_{i,j}$). Since we deal with an unsupervised classification task, we aim to separate the involved DHB pages into 2 clusters (*i.e.* to group pages by type of layout and/or content to separate the most common or frequent pages which have similar layout and/or content from those that have particular layout and/or content). One cluster representing frequent pages having similar layout and/or content and the other one illustrating pages having particular layout and/or content.

7.4.2. Page stream segmentation

Secondly, by only analyzing the $m^g_{i,i+1}$ elements of M^g, the different pairs of the successive DHB pages can be grouped or retrieved according to a pre-defined threshold GED value. This task aims to retrieve the transition pages in the involved DHB (*i.e.* identify different series of successive pages having distinct layout and/or content). It is worth noting that this task is considerably important, since it can detect pages having scanning failure occurring during the digitization process (e.g. blur, skewed or folded pages). Moreover, by identifying these transition pages (e.g. title pages of chapter), a particular indexing process can be carried out to assist a user in generating a table of contents/summary (*i.e.* DHB page stream segmentation).

7.5. Experiments and results

In this section, the experimental protocol is firstly described. Then, qualitative results and an assessment of the different steps of the proposed approach used to generate this signature are presented. Subsequently, an analysis of the obtained results is discussed. Afterwards, a thorough evaluation has been conducted for assessing two possible signature-based applications, unsupervised book page classification and book page stream segmentation, to illustrate the potential of the proposed signature.

7.5.1. Experimental protocol

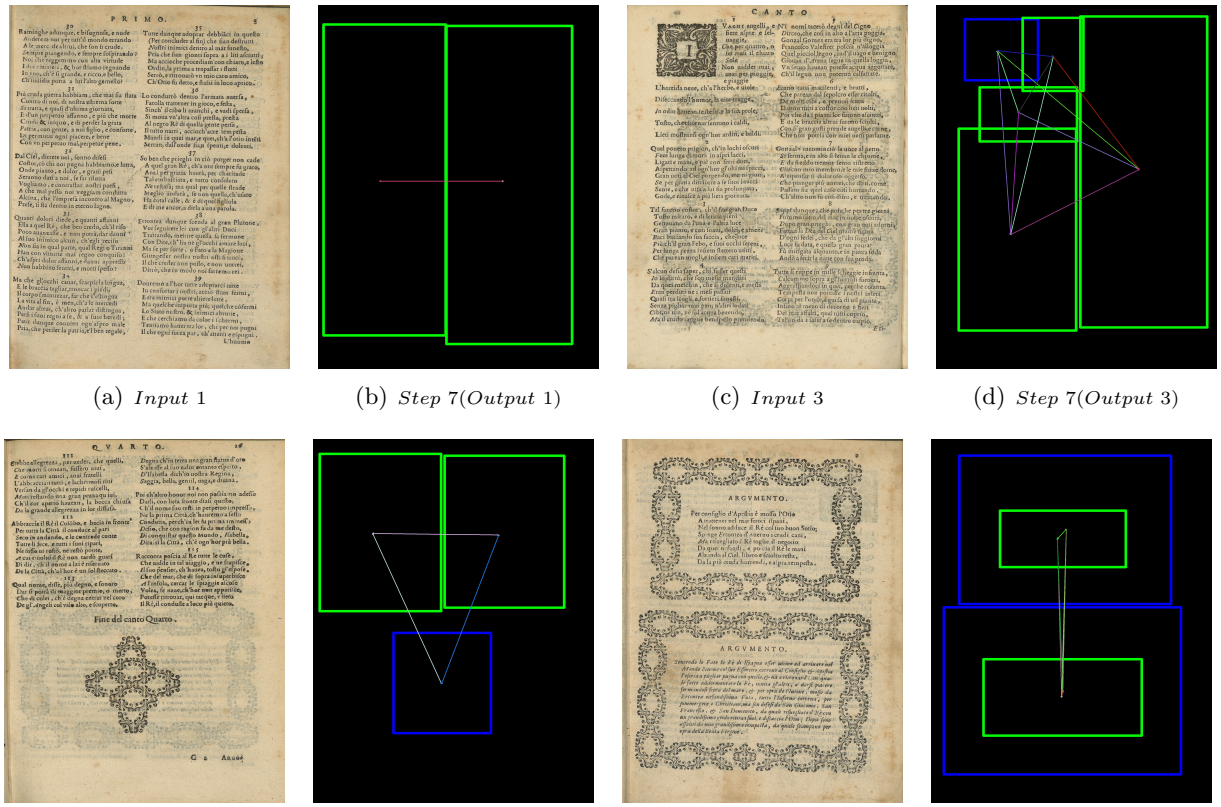
Since the proposed structural signature based on texture for book page characterization and categorization is used on an entire book (*i.e.* all pages of the DHB under consideration), our experimental corpus in this chapter contains one DHB (a printed monograph which is dated 1596, titled "Il mondo nuovo, del sig. Giov. Giorgini da Jesi" and written in Italian) which is composed of 322 ground-truthed one-page color HDIs[3]. The analyzed DHB has been collected from Gallica[3], and its pages have been digitized at 300 dpi and saved in the TIFF format. The analyzed DHB consists of 81 pages containing graphical and textual regions (*i.e.* pages that have particular layout and/or content) and 241 pages containing only textual regions (*i.e.* the most common or frequent pages that have similar layout and/or content).

Due to the constraints of the timelines for submitting this dissertation, the proposed signature for DHB page categorization has been analyzed and evaluated on one DHB. Nevertheless, the assessment of the proposed DHB page signature will be completed by extending our experiments to nine other DHBs which is already ongoing work, *i.e.* 4372 gray-scale/color manuscript/printed pages which encompass six centuries of French history (1201-1822) are currently being investigated.

[3] `http://gallica.bnf.fr/ark:/12148/bpt6k132294p/f5.planchecontact.r=.langFR`

7.5.2. Characterization of digitized historical book pages

To evaluate the performance of the proposed signature-based approach for DHB page characterization, qualitative and quantitative evaluation of the different steps of its extraction are presented in Figure 7.5 and Table 7.2, respectively.



(a) *Input* 1 (b) *Step* 7(*Output* 1) (c) *Input* 3 (d) *Step* 7(*Output* 3)

(e) *Input* 4 (f) *Step* 7(*Output* 4) (g) *Input* 2 (h) *Step* 7(*Output* 2)

Figure 7.5.: Illustration of the resulting HDIs derived from the proposed approach for DHB page characterization using the proposed graph-based signature.

Table 7.2.: Evaluation of the different steps of the proposed approach for DHB page characterization.

		***Step* 3**	***Step* 4**	***Step* 5**
CA	μ	0.977	0.983	0.987
	σ	0.066	0.085	0.076

		***Step* 6**
J_{AR}	μ	0.952
	σ	0.174

The success of the proposed approach is demonstrated by visual inspection of the segmented HDIs (*i.e.* homogeneous regions are determined by identifying the graphic regions (blue) and textual regions (green)). Then, the pixel-based classification accuracy (CA) is computed to evaluate quantitatively the obtained results of the following steps of the proposed approach for DHB page characterization: the pixel-clustering and labeling step (*Step* 3), the pixel-labeling refinement step (*Step* 4) and the post-processing step (*Step* 5). Another accuracy metric is calculated, the Jaccard index (J_{AR}), for assessing the step of extraction of representative homogeneous regions (*Step* 6) [469]. $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of (.), respectively. The higher the mean values, the better the results. High performances of the computed accuracy metrics are obtained for the different steps of the proposed signature-based approach (*i.e.* more than 95%). Moreover, a slight gain in the average value of CA is obtained from one step to the next, in order to achieve the aim of identifying homogeneous regions with an average value of J_{AR} equal to 95%.

7.5.3. Categorization of digitized historical book pages

A GUI tool for characterization and categorization of DHB pages is designed in this work to illustrate graphically the performance of different signature-based applications (e.g. unsupervised DHB page classification and DHB page stream segmentation). It has been developed using the C++ language, openCV library and Qt development environment.

First, we can see in Figure 7.6(a) the separation of the DHB pages into 2 clusters. One cluster representing frequent pages having similar layout and/or content and the other one illustrating pages having particular layout and/or content. Each cluster is represented in a separate line. In our experiments and particularly in the DHB on which the evaluation has been performed, we note that the clustering achieves a distinction between pages having similar layout and/or content (*i.e.* double columns of text), and those having particular layout and/or content (*i.e.* textual and graphical regions).

Second, we can see in Figure 7.6(b) the different detected transition DHB pages. Only DHB pages having GEDs above a pre-defined threshold GED value are retrieved. The shaded DHB pages are considered as non-transition pages, while the DHB pages with red borders are considered as the transition pages (*i.e.* they have layout and/or content that differ from the following page). Using the developed computer-aided tool for characterization and categorization of DHB pages in this work, users are able to vary the threshold GED in order to increase or decrease the number of transition pages. The proposed tool for characterization and categorization of DHB pages provides an integrated user-centered GUI which is specifically engineered to make it easy the identification of the transition pages in the DHB under consideration according to the user requirements.

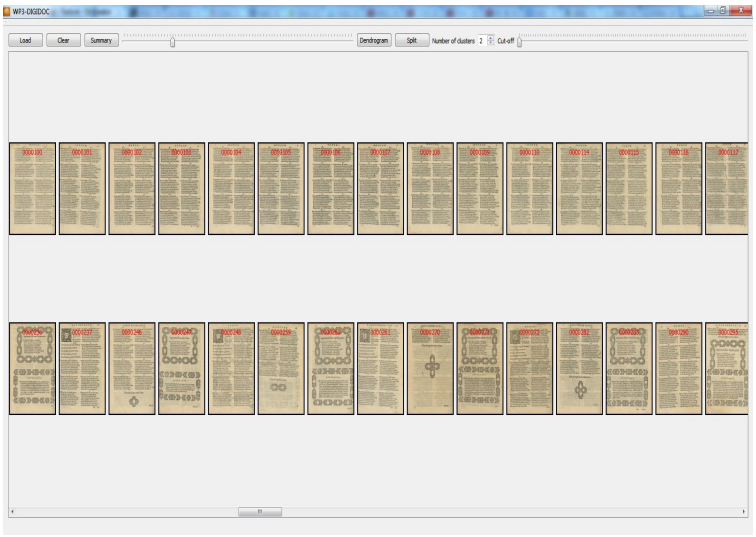
In Appendix B and particularly in Section B.10, other screen shots of the designed computer-aided tool are illustrated for characterization and categorization of DHB pages.

7.5.3.1. Unsupervised page classification

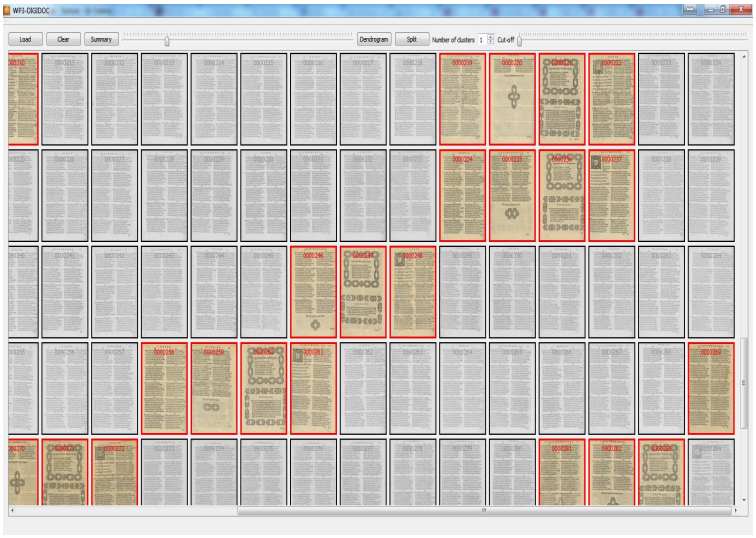
To get an insight into the classification accuracy, a confusion matrix is computed (*cf.* Table 7.3). The confusion matrix illustrates one cluster containing the most common pages in the involved DHB (*i.e.* pages containing only text) on the one hand, and those considered as particular pages in the involved DHB (*i.e.* pages containing text and graphics) on the other hand. The following classification accuracy measures are computed: precision (P), recall (R), F-measure (F) and classification accuracy (CA). P_i and R_j denote the individual cluster precision and recall, respectively. For the cluster representing the most common pages in the involved DHB (*i.e.* pages containing only text), 91%(P) and 94%(R) are obtained. On the other side, for the cluster representing the pages that have particular layout and/or content (*i.e.* pages containing text and graphics), we find 85%(P) and 77%(R). Thus, we show that the proposed approach tends to miss-classify more the pages containing textual and graphical regions than those containing only textual regions, due to the complexity of the layout and content of the particular DHB pages (*cf.* Figure 7.5(g)). Nevertheless, the overall result is quite encouraging, since we obtain 87%(F) and 90%(CA). This confirms that the proposed signature ensures the unsupervised DHB page classification according to the DHB page layout and content.

Table 7.3.: Evaluation of the proposed signature for unsupervised DHB page classification.

		Ground-truth		
		Class 1	**Class 2**	
Clustering outcomes	**Cluster 1**	69	12	$\leftrightarrow \boldsymbol{P_1 = 0.85}$
	Cluster 2	20	221	$\leftrightarrow \boldsymbol{P_2 = 0.91}$
		$\updownarrow$	$\updownarrow$	
		$\boldsymbol{R_1 = 0.77}$	$\boldsymbol{R_2 = 0.94}$	



(a) Unsupervised page classification



(b) Page stream segmentation

Figure 7.6.: Screen shots illustrating graphically the performance of the two analyzed and evaluated signature-based applications.

7.5.3.2. Page stream segmentation

By analyzing the $m^g_{i,i+1}$ elements of the normalized distance matrix M^g, the different pairs of the successive DHB pages can be grouped according to different GED values. As a matter of fact, the pairs of the successive DHB pages that have lower GED values, have certainly similar layout and/or content (*i.e.* non-transition pages). On the other side, the other pairs that have higher GED values correspond to pages have different layout and/or content (*i.e.* transition pages such as the title pages of chapter). By analyzing the composition of the involved DHB, 102 pairs of the successive DHB pages are identified as pairs of transition pages. From these pairs, 128 DHB pages are considered as transition pages. By drawing the histogram of the computed GED values between each pair of successive DHB Pages (*cf.* Figure 7.7), on peak is showed with lower values of GED (*i.e.* the lower the GED value, the more similar the pages in terms of layout and content). This histogram peak corresponds to the number of detected pairs of successive DHB Pages which have similar layout and/or content (*i.e.* double columns of text) and can be identified as non-transition pages according to the obtained GED value. Thus, this confirms that the proposed graph-based

signature used for page stream segmentation is robust and relevant.

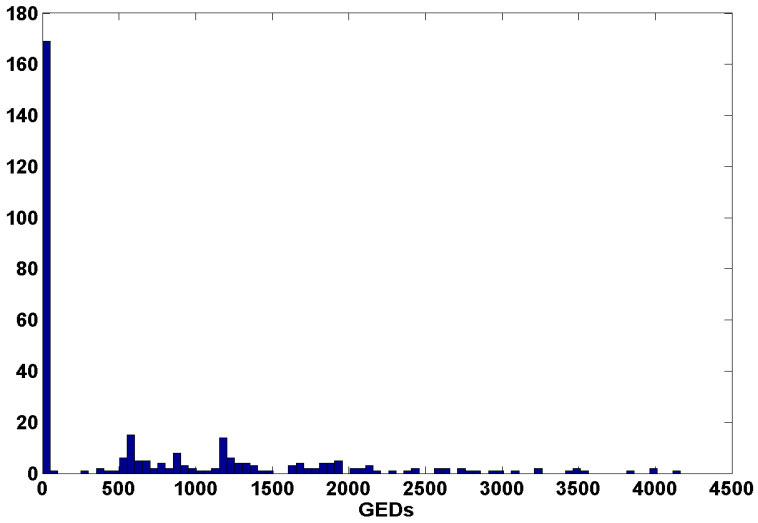


Figure 7.7.: Histogram of the computed GED values between each pair of successive DHB Pages for DHB page stream segmentation.

Figure 7.8 illustrates the ROC curve by varying the GED threshold values illustrating the good performance of the proposed signature for the identification of the transition DHB pages. This strengthens our previous results and confirms that the proposed signature ensures the identification of the transition pages in a DHB such as the title pages of chapter and subsequently it allows the DHB page stream segmentation.

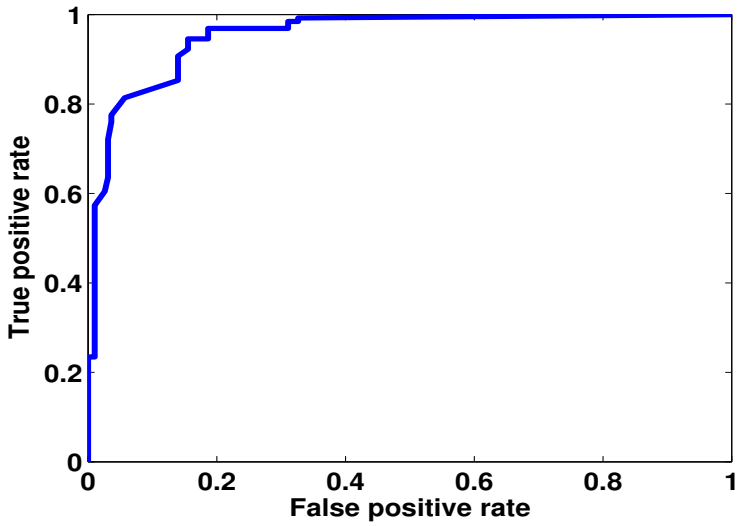


Figure 7.8.: Evaluation of the proposed page signature for DHB page stream segmentation.

7.6. Discussion

The first aspect of future work will be to use the proposed signature on a larger corpus. This study is ongoing and will evaluate the signature more adequately, with more convincing experimental results.

Then, we will assess other possible applications of the proposed graph-based signature:

- Finding pages in a DHB or HDI corpus which contain a particular content component or a group of patterns that match specific criteria defined by a user (*i.e.* investigating the sub-graph isomorphism paradigm).

- Retrieving similar pages in a HDI corpus query tool by establishing a ranking based on the computed GEDs between the corpus pages and the query page. This ranking can be adjusted automatically according to the weights of each category of the computed features in the cost of the edit operations when performing the GED. This will ensure that either the layout structure (e.g. topological, geometric and shape attributes) or the typographic/graphical characteristics of content (e.g. texture attributes) of the HDIs under consideration can be highlighted.

- Detecting the scanning failure occurring during the digitization process (e.g. curvature, light) to ensure effective computed-aided quality control of the digitization.

Furthermore, we will investigate a finer unsupervised book page classification with different values of the number of clusters. We also intend to analyze the impact of different feature weighting schemes in the cost of the edit operations when computing the GED. In addition, further work also needs to compare the results given by using the approximate GED computed on the involved graph-based DHB page signatures with other state-of-the-art graph dissimilarity techniques.

Finally, improvements can be made regarding the designed GUI tool for characterization and categorization of DHB pages. In particular, advanced human-computer interaction techniques can be introduced to optimize the way in which the users interact with scanners during the digitization process.

7.7. Conclusion

Since the ultimate goal of the DIGIDOC project is developing relevant ways of interacting with scanners by assisting the digitization operator to adjust automatically the best set of parameters (e.g. resolution, lightening, color calibration), detecting errors in the digitization process (e.g. blur, skewed or folded pages), providing appropriate assistance for document indexing (e.g. by recognizing automatically page types or breaks in a sequence of pages), *etc.*, a simple GUI tool for characterization and categorization of DHB pages is designed. The designed tool proposes an integrated user-centered GUI which is specifically engineered to make it easy the identification of the transition or similar layout and/or content pages in the DHB under consideration according to the user requirements. The GUI tool is based on a generic graph-based signature for DHB page characterization and categorization.

The proposed graph-based signature is generated for each DHB page based on characterizing each DHB page with a set of homogeneous texture regions with varying low-level features. The proposed structural signature ensures the characterization of the DHB page layout and content. In addition, this signature guarantees the implementation of numerous applications for managing effectively a corpus or collections of books (e.g. information retrieval in digital libraries according to several criteria or page categorization). As a consequence, by comparing the different graph-based signatures, the DHB pages with similar layout and/or content pages can be grouped. A thorough evaluation has been conducted in this work for assessing two possible applications of the proposed signature, unsupervised book page classification and book page stream segmentation, and it has achieved promising results.

The proposed signature ensures firstly a relevant unsupervised DHB page classification according to the DHB page layout and content. 90% classification accuracy is noted for the first signature-based application, unsupervised DHB page classification. Then, the proposed signature has the ability to identify the transition pages in a DHB such as the title pages of chapter and subsequently it allows the DHB page stream segmentation. Encouraging results are observed for the second

signature-based application, DHB page stream segmentation. Hence, a table of contents/summary of the DHB under consideration has been automatically generated by detecting different or dissimilar pages. This will ensure fast and easy navigation on historical collections on the one hand and effective computed-aided quality control of the digitization (e.g. detecting the scanning failure occurring during the digitization process) on the other hand.

Chapter 8.

Conclusions and future perspectives

This chapter summarizes some conclusions about the work presented in this dissertation and possible future directions in historical document image analysis.

Contents

Throughout this dissertation, several methods and a number of studies for historical DIA and DHB page characterization have been presented. This chapter summarizes the work presented in this dissertation by revisiting the contributions, strengths and weaknesses. Finally, an overview of the future research possibilities in the area of historical DIA is discussed.

8.1. Conclusions and contributions

This section briefly summarizes the conclusions and contributions of this dissertation.

8.1.1. Conclusions

In this dissertation, we have presented six chapters.

1. First, we have summarized the research projects related to digital libraries and historical DIA in Chapter 2.

2. Then, we have detailed related works on DIA in Chapter 3, by reviewing the classical and texture-based approaches, with a particular focus on those related to historical DIA.

3. Chapter 4 has presented an experimental evaluation and benchmarking of nine evaluated texture-based feature sets (Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor, Haar, Db3 and Db4). This comparative study has been conducted on a large corpus of HDIs for the purpose of determining the performance of each texture-based feature set according to the DI content, *i.e.* segmenting graphical regions from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand. Using a standard pixel-labeling scheme for evaluating and benchmarking texture features, we have shown the scalability for two datasets, the *"DIGIDOC-Texture dataset"* and *"HBR2013 dataset"* (1100 pages of historical documents). This work has shown the effectiveness of the texture analysis approaches for historical DIA. Based on our experiments, we conclude that the auto-correlation, Gabor and Db4 features are the best choices for discriminating textual regions from graphical ones without taking into account the spatial relationships between pixels. However, when the numerical complexity and pixel-labeling performance are taken into account, the Gabor approach would be the better choice. Furthermore, the Gabor approach is a good choice for segmenting HDIs containing only textual regions with different fonts. 76%, 80% and 76% classification accuracy values are noted when the auto-correlation, Gabor and Db4 are used in the proposed pixel-labeling scheme for evaluating and benchmarking texture features, respectively. The results reported in this chapter provide a useful benchmark in terms of performance evaluation, texture vector dimensionality, memory requirements, processing time and complexity for current and future research efforts in historical DIA.

4. Chapter 5 has proposed a generic framework for a texture-based pixel-labeling framework of DHB content with no hypothesis concerning the document layout or the typographic/graphical characteristics of the document. The aim of this framework is to group pixels having similar DHB page content type within the content of DHBs by extracting and analyzing texture features independently of the layout of the pages. It is therefore applicable to a large variety of books. The proposed framework is based on a feature vector that is composed of texture indices. Texture features are extracted from the different areas of a page and at several resolutions. The robustness of the extracted features is used in a parameter-free unsupervised clustering method which is performed to determine the number of book content types (*i.e.* defined by similar texture indices). Moreover, the number of book content types does not need to be known in advance as it is automatically determined. The proposed framework has been evaluated on the *"DIGIDOC-Framework dataset"* which is composed of 316 pages of HDIs. We conclude that texture features provide a good discrimination of the foreground

layers of DHB pages, particularly between text and graphics. 85% purity per block accuracy and 79% classification accuracy are obtained for the auto-correlation-based framework, while 89% purity per block accuracy and 77% classification accuracy are noted for the Gabor-based framework.

5. Chapter 6 has described an automatic characterization approach of DHB pages. The characterization is embedded in what we call a structural signature of DI. Generating a structural signature for each analyzed DHB page is carried out in three stages: the first step consists in refining the obtained pixel-labeling results by taking into account the topological or spatial relationships between pixels, the second one aims to extract homogeneous regions and the third one is generating a graph-based signature of the page content and structure. The proposed signature does not assume *a priori* knowledge regarding page layout and content, and hence, it is applicable to a large variety of ancient books. By integrating varying low-level features (e.g. texture) characterizing the different page components (different text fonts and graphic regions) on the one hand, and structural information describing the page layout on the other hand, the proposed signature provides a rich and holistic description of the layout and content of the analyzed book pages. The proposed characterization approach of DHB pages gives encouraging results since 77% of Jaccard index is noted when we have evaluated the extracted homogeneous regions.

6. Chapter 7 has illustrated the effectiveness of the proposed page signature. By conducting a thorough experimental evaluation in the context of the DIGIDOC project, two possible signature-based applications, unsupervised page classification and page stream segmentation, have been assessed with the aim of managing effectively a corpus or collections of books. Hence, by comparing the different graph-based signatures, the involved DHB pages with similar layout and/or content pages can be grouped. As a consequence, the proposed signature ensures firstly a relevant unsupervised DHB page classification according to the DHB page layout and content. 90% classification accuracy is noted for the first signature-based application, unsupervised DHB page classification. Then, the proposed signature has also the ability to identify the transition pages in a DHB such as the title pages of chapter, and subsequently it allows the DHB page stream segmentation. Encouraging results are observed for the second signature-based application, DHB page stream segmentation. To illustrate the potential of the proposed graph-based signature, a simple GUI tool for characterization and categorization of DHB pages is designed. The designed tool proposes a simple integrated user-centered GUI which is specifically engineered to make it easy the identification of the transition or similar layout and/or content pages in the DHB under consideration according to the user requirements.

8.1.2. Contributions

This dissertation has made a number of contributions towards the goal of designing a computer-aided characterization and categorization tool of HDIs, able to index or group DHB pages according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the HDI content. Key contributions of this work are:

1. Presenting an experimental evaluation and benchmarking of a number of commonly and widely used texture features which have been conducted on a large corpus of HDIs for the purpose of determining the performance of each texture-based feature set according to the DI content, *i.e.* segmenting graphical regions from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand.

2. Proposing a texture-based pixel-labeling framework that is used on an entire book instead of processing each page individually, for the segmentation and analysis of DHB content. The

proposed framework is supported by the fact that pages of the same book usually present strong similarities in the organization of the HDI information (*i.e.* layout) and in the graphical and typographical features (*i.e.* content) throughout the DHB pages under consideration.

3. Defining a structural representation based on texture which is called a graph-based signature, for DHB page characterization. The proposed signature is based on varying low-level features (*i.e.* texture, shape, geometric and topological descriptors) and a structural signature. It provides a topological signature of digitized historical book page according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the HDI content.

4. Illustrating the effectiveness of the proposed page signature by firstly conducting a detailed experimental evaluation for assessing two possible signature-based applications, unsupervised page classification and page stream segmentation, secondly by designing a simple integrated user-centered GUI which is specifically engineered to make it easy the identification of the transition or similar layout and/or content pages in the DHB under consideration according to the user requirements.

8.2. Future perspectives

There are many directions to proceed in the work presented in this dissertation.

The first aspect of future work will be to use the proposed methods and studies in this dissertation on a larger database. This is ongoing and will evaluate the different studies and proposed methods more adequately with more convincing experimental results in order to help improve their scalability. We will then study and combine statistical, geometric, model-based and spectral texture-based features in order to refine the segmentation and ensure a distinction between different text fonts and various graphic types.

Historical DIA is still an open issue for both supervised and unsupervised methods due to the variability of the contents and/or layouts of historical documents. As for the supervised methods, feature learning or representation learning [529] will be investigated for pixel-classification in future research. This helps in dealing with retrieving relevant features or representations from raw data. In addition, a feature selection step (e.g. dimension reduction technique) can also be integrated to select relevant features and remove redundant ones.

Concerning the proposed approach for texture feature extraction based on multi-scale analysis, we propose to introduce the superpixel approach [530, 531] into the texture feature analysis step. The superpixel approach becomes a consistent alternative of using a rigid structure of pixel grid, *i.e.* it is faster, it has a lighter memory consumption, and it is more interesting to compute image features on each superpixel center than on each image pixel.

In order to assess the robustness of the proposed texture-based approaches, images of historical documents under numerous degradation models will be generated and image enhancement algorithms (e.g. non-local means filtering [532] and total variation [533]) will be integrated. This study will show the robustness of texture feature extraction for segmentation in the case of noise and the uselessness of a denoising step.

In this work, a generic signature for DHB page characterization and categorization has been evaluated on two possible applications, unsupervised book page classification and book page stream segmentation, with no hypothesis concerning page layout and content. Then, we will assess other possible applications of the proposed graph-based signature:

- Finding pages in a DHB or HDI corpus which contain a particular content component or a group of patterns that match specific criteria defined by a user (*i.e.* investigating the sub-graph isomorphism paradigm).
- Retrieving similar pages in a HDI corpus query tool by establishing a ranking based on the computed GEDs between the corpus pages and the query page. This ranking can be adjusted automatically according to the weights of each category of the computed features in the cost of the edit operations when performing the GED. This will ensure that either the layout structure (e.g. topological, geometric and shape attributes) or the typographic/graphical characteristics of content (e.g. texture attributes) of the HDIs under consideration can be highlighted.
- Detecting the scanning failure occurring during the digitization process (e.g. curvature, light) to ensure effective computed-aided quality control of the digitization.

Furthermore, we will investigate a finer unsupervised book page classification with different values of the number of clusters. We also intend to analyze the impact of different feature weighting schemes in the cost of the edit operations when computing the GED. In addition, further work also needs to compare the results given by using the approximate GED computed on the involved graph-based DHB page signatures with other state-of-the-art graph dissimilarity techniques.

In addition, we will then focus on demonstrating the robustness of the proposed solutions and provide additional insights into their accuracies by investigating and analyzing parts of historical document images (e.g. handwritten annotations) or graphic images (e.g. illustrations and drop caps).

Moreover, improvements can be made regarding the designed GUI tool for characterization and categorization of DHB pages. In particular, advanced human-computer interaction techniques can be introduced to optimize the way in which the users interact with scanners during the digitization process.

Finally, a public annotated dataset of HDIs will be available soon to initiate collaborative research but a larger pixel-based ground-truth is needed to be more subjective and fully representative of the diversity of HDIs, to train algorithms and to evaluate research works related to historical DIA. Our future work will also focusing on analyzing four other state-of-the-art ground-truthing tools, TrueViz[1], WebGT[2], Aletheia[3] and Divadia[71] for more reliable performance evaluations.

Appendices

Appendix A.

Related works

Contents

A.1. Feature space structuring methods in the literature

The feature space structuring methods aim to partition and analyze the set of unlabeled data into groups or clusters. They involve two phases:

- ***Clustering phase*** or unsupervised classification partitions a set of unlabeled data into homogeneous groups or clusters. Samples of each cluster share common characteristics which usually correspond to proximity criteria, defined by introducing measures of distance between clusters and samples.
- ***Classification phase*** classifies a new object according to a set of pre-defined classes.

Clustering algorithms can be classified into two categories:

- ***Hard clustering methods*** distribute data into different clusters, where each data point belongs to exactly one cluster.
- ***Fuzzy clustering methods*** consider that the allocation of data points to clusters is not binary, *i.e.* each data point may belong to more than one cluster with a set of membership levels. One of the most widely used fuzzy clustering algorithms is the FCM method [534].

In this work, we are interested in the hard clustering algorithms since many parameters must be specified in the case of the fuzzy clustering methods. Several standard hard clustering methods have been proposed in the literature. Hard clustering methods are divided into five categories [535]:

- ***Partitioning methods*** (e.g. k-means clustering (k-means) [331], partitioning around medoids (PAM) [176], Clustering large applications (CLARA) [176]) distribute the dataset according to the proximities of feature space deducted from the content of the analyzed image.
- ***Hierarchical methods*** (e.g. agglomerative nesting (AGNES) [176], divisive analysis clustering (DIANA) [176], hierarchical agglomerative clustering (HAC) [332]) are widely used data analysis tools that produce a hierarchy of clusters based on a measure of similarity between groups of data points.
- ***Density-based methods*** (e.g. DBSCAN [536], OPTICS [537], expectation-maximization (EM) algorithm [538]) are designed to reveal clusters of arbitrary shapes based on the local densities of a point set after introducing the appropriate values of the input parameters (neighborhood radius, *etc.*).
- ***Grid-based methods*** (e.g. STING [539], WaveCluster [540]) quantize the space into a finite number of cells without taking into consideration data density and distribution and then perform clustering operations (neighborhood cells, *etc.*) on the quantized space.
- ***Neural network-based methods*** (e.g. self-organizing maps (SOM) [541], feed-forward network (FFN) [542]) partition data into similar sub-sets with the help of an artificial neural network [543].

The different feature space structuring techniques that have been used with HDIs are summarized in Table A.1.

Table A.1.: Clustering algorithms used with HDIs in the literature.

Ref.	Data kind	Algorithm class/Number of clusters	Clustering algorithm	Description
[1]	Entire books (printed)	-Unsupervised -The number of clusters was assumed to be known in advance	CLARA	The non-supervised clustering technique was applied on extracted texture features which were computed from six pages of the same book.
[4, 57]	Entire gray-scale or color pages (handwritten historical manuscripts)	-Supervised -The algorithm required knowing the number of classes in advance	SVM	A physical structure detection method for historical handwritten DIs was proposed by classifying and labeling each pixel as periphery, background, text block or decoration using SVM.
[89, 229]	Entire color high resolution digitized images (manuscripts and printed)	-Supervised -The number of clusters was assumed to be known in advance	SVM and radial basis function as kernel	Text, images and their associated captions were extracted using a SVM classifier trained on the extracted texture features.
[29]	Drop caps (printed)	-Unsupervised -The number of clusters was assumed to be known in advance	k-means	The clustering technique was used on computed texture descriptors.
[53, 55]	Entire gray-scale or color pages (handwritten historical manuscripts)	-Supervised	Dynamic MLP (DMLP)	Multi-resolution physical layout analysis and segmentation of medieval manuscripts with three analysis levels using a series of images with increasing resolution. The classification on each level was performed by DMLP classifier on the pixel color features and pixel positions extracted from the scaled HDIs. A manual annotation was needed for producing a training set.
[56]	Entire gray-scale or color pages (handwritten historical manuscripts)	-Supervised -The algorithm required knowing the number of classes in advance ($k = 4$)	SVM, MLP and Gaussian mixture models (GMM)	Comparison between three classifiers based on SVM, MLP and GMM was firstly performed to detect physical structure of HDIs. Pixels were classified into 4 classes: periphery, background, text or decoration, in the first classification level. Then, the three evaluated classifiers were combined together to ensure a vote for the pixel label in order to further improve the pixel-labeling results. They concluded that both SVM and MLP classifiers had better performance than GMM.
[245]	Entire pages (manuscripts)	-Supervised -The number of clusters was assumed to be known in advance	SVM and radial basis function as kernel	A manually given annotated training dataset and a radial basis function were used with an embedding procedure for SVM classification.
[275]	Entire pages (MadiaTeam document database)	-Unsupervised -The algorithm required knowing the number of classes in advance ($k = 3$)	k-means	Grouping same content blocks using the k-means clustering. The three defined classes present in documents were, text, graphics and space.
[237]	Collected character images and pages of word set (printed)	Fuzzy methods	Fuzzy membership functions was built from fuzzy logic and fuzzy set theory	OCR based on training step with collected character image examples with the help of fuzzy membership function.
[238]	Entire pages (printed periodicals)	-Unsupervised -The number of clusters was assumed to be known in advance	k-means	The k-means algorithm was used to separate text, background and image.
[239]	Image patches	-Supervised -The number of clusters was assumed to be known in advance	Random forest classifier	The supervised learning technique was used on computed texture descriptors obtained by using dimensionally reduced multi-channel GFs for selecting informative features.

Table A.1 – continued from previous page

Ref.	Data kind	Algorithm class/Number of clusters	Clustering algorithm	Description
[242]	Entire pages (printed and manuscripts)	-Unsupervised -The algorithm required knowing the number of classes in advance ($k = 3$)	k-means	Text, background and image were separated using the k-means algorithm.
. [541]	Printed book (Gutenberg Bible)	-Supervised -Each pixel was classified into one class out of four in the testing step	FFN and SOM	The SOM algorithm was trained to generate a set of test vectors that was used in the FFN algorithm.
[330]	Drop caps (printed)	-Unsupervised -The number of clusters was determined automatically	HAC (inconsistency criterion)	The HAC algorithm was used on texture features to classify the drop caps strokes.
[243]	Entire pages (degraded official administrative documents)	-Unsupervised -The number of clusters was assumed to be known in advance	k-means	Segmentation of complex multi-lingual multi-script documents: separation text/graphics and extraction of graphs, tables and text lines.
[244]	Entire pages (manuscripts)	-Unsupervised -The algorithm required knowing the number of classes in advance	FCM	Text/graphics segmentation using the FCM algorithm. The clustering approach generated classes based on the sum of squared deviations inter-class and intra-class.
[544]	Ancient books (printed and manuscripts) and archival materials	-Supervised (training and self-learning stages)	Not mentioned	Segmentation, recognition and transcription of text in a set of digital images referring to pages of ancient manuscripts or printed books.
[545]	Entire pages (manuscripts)	-Unsupervised -The initial parameters of the EM were estimated by the k-means algorithm	k-means and EM	Text extraction algorithm from degraded documents on the basis of the probabilistic models.
[546]	Text documents	-Unsupervised	Bayes criteria	The Font classification step was performed on fractal descriptors which were calculated from extracted local text zones.
[547]	Entire pages (color manuscripts)	-Unsupervised -The user intervened in the initialization step by defining the different samples of colors for each class and the number of classes	Serialization of the k-means algorithm	Serialized classifier was applied for adaptive color image segmentation for digitized ancient manuscripts.

A.2. Clustering and classification accuracy metrics in the literature

The performance evaluation idea consists of quantifying how the clusters given by a clustering technique are different from the classes defined in the ground-truth, as shown in Figure A.1.

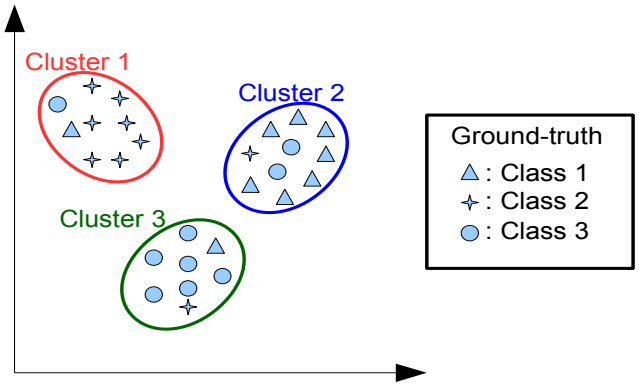


Figure A.1.: Clustering result *vs.* ground-truth.

A.2.1. Clustering accuracy metrics

First, to evaluate a segmentation/classification approach, several clustering accuracy metrics have been proposed [340, 548, 549]. General segmentation method evaluation surveys have been presented in the literature [340, 548, 549]. The clustering accuracy metrics are classified into two kinds [550, 551, 334] (*cf.* Table A.3):

A.2.1.1. Internal or unsupervised measures

The internal or unsupervised measures evaluate the clustering quality by considering only the intrinsic information concerning the distribution of the observations into different clusters (e.g. silhouette width index (SW) [341], Dunn index [397], Davies-Bouldin index [391], compactness [535], homogeneity [552]). They often assess the clustering result based on the two criteria: the compactness and separation. The compactness measures how closely the points in a cluster are, while the separation quantifies how separate different clusters are. For example, the SW measures the level of compactness and separation by analyzing the distribution of the observations into clusters. The silhouette width $SW(x_i)$ for each point x_i estimates how much x_i belongs to its cluster. It is computed as follows:

$$SW(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}} \tag{A.1}$$

where $a(x_i)$ and $b(x_i)$ represent the compactness between x_i and the other points in the same cluster and the separation between x_i and the closest cluster, respectively. $a(x_i)$ is obtained by computing the average distance between x_i and the other points in the same cluster. On the other hand, $b(x_i)$ is given by calculating the average distance between x_i and the points in another cluster that does not contain the point x_i and which is the closest to x_i. $a(x_i)$ and $b(x_i)$ are computed as follows:

$$a(x_i) = \frac{1}{|K(x_i)|-1} \sum_{x_j \in K(x_i), x_j \neq x_i} D(x_i, x_j) \tag{A.2}$$

where $K(x_i)$ represents the cluster containing the point x_i. $D(x_i, x_j)$ is the distance between two points x_i and x_j.

$$b(x_i) = \min_{K_l \neq K(x_i)} \{D(x_i, K_l)\} \tag{A.3}$$

where K_l is the cluster that does not contain the point x_i. $D(x_i, K_l)$ is defined as:

$$D(x_i, K_l) = \frac{1}{N_l} \sum_{x_j \in K_l} D(x_i, x_j) \tag{A.4}$$

where N_l is the number of points in the cluster K_l.

Finally, to evaluate the quality of the clustering result, the average silhouette width of all points in the dataset (SW). The higher the values, the better the results. SW is defined as:

$$SW = \frac{1}{N} \sum_{x_i \in X} SW(x_i) \tag{A.5}$$

where N is the number of points in the dataset.

A.2.1.2. External or supervised measures

The external or supervised measures compare the distributions of the observations in the clustering result and ground-truth (e.g. rand index [401], Jaccard coefficient (J) [342], Fowlkes-Mallows index [405]). They often compare the clustering result with the ground-truth using the two following criteria: the homogeneity and completeness. The homogeneity criterion of a clustering result is satisfied if all obtained clusters contain only points of a single class. On the other hand, the completeness criterion is satisfied if all points which belong to a single class of the ground-truth are assigned to a single cluster. For example, the J is used to assess the similarity between the distributions of the observations in the clustering result and ground-truth. It represents the ratio of the number of pairs of data points which are clustered similarly in the clustering result and ground-truth. The value of the J ranges between $[0, 1]$. The higher the values, the better the results. J is defined as:

$$J = \frac{N_{11}}{N_{11} + N_{10} + N_{01}} \tag{A.6}$$

where N_{11}, N_{10} and N_{01} represent the number of pairs of data points which are clustered together in the clustering result and ground-truth which are clustered together in the clustering result but not in the ground-truth and which are clustered together in the ground-truth but not in the clustering result, respectively.

A.2.2. Classification accuracy metrics

Then, in order to provide an additional analysis and comparison with the computed clustering accuracy metrics and get an insight into the classification accuracy, a confusion matrix, error matrix or contingency table (M_c) is computed [343, 344]. From the M_c, several classification accuracy metrics are deduced, including entropy (E), purity (PT), precision (P), recall (R), classification accuracy rate (CA) and F-score or F-measure (F) (*cf.* Table A.3) [345, 346, 347, 348, 349]. These accuracy metrics are related to how representative the clusters are of classes and help to determine classes which are not able to segregate groups of data and give an insight into the confusion and misclassification rates. They help us to determine how good the clustering is and compare the cluster memberships with the class ones [348]. As evaluation clustering criterion, we assume that preference will be given to higher PT, P, R, CA and F and to lower E.

Let the set of classes in the dataset be $c = \{c_1, c_2, ..., c_i, ..., c_k\}$. $Pr_i(c_j)$ is the proportion of the data point class c_j in the cluster i.

The PT of a cluster corresponds to the ratio of the largest class of pixels assigned to this cluster with respect to the overall cluster size. It is computed with:

$$PT = \max_{j}(Pr_i(c_j)) \tag{A.7}$$

The E evaluates the confusion level through the M_c from the class distribution of misclassified pixels. It is defined as follows:

$$E = -\sum_{j=1}^{k} Pr_i(c_j)\log_2(Pr_i(c_j)) \tag{A.8}$$

In this work, the M_c columns correspond to the reference data, assumed as our defined ground-truth, and the M_c rows correspond to the clustering result. The diagonal elements of the M_c represent the all correctly assigned samples to theirs classes. Therefore, an absolutely correct clustering will result in a diagonal matrix. The elements of the M_c, excluding those of its diagonal, along a column (clustering outcomes) correspond to the omission samples, *i.e.* the reference elements from one class are assigned to another one. On the other hand, the elements of the M_c, excluding those of its diagonal, along a row (ground-truth classes) correspond to the commission samples, *i.e.* in the opposite case where the samples of a class assigned to it by mistake.

First, we define N_d as the number of the M_c diagonal elements which represent the all correctly assigned samples to theirs classes. Then, N_o denotes the number of the M_c elements, excluding those of its diagonal, along a column (clustering outcomes) correspond to omission samples. Finally, N_c represents the number of the M_c elements, excluding those of its diagonal, along a row (ground-truth classes) correspond to commission samples.

Thus, the P is given by:

$$P = \frac{N_d}{N_d + N_c} \tag{A.9}$$

The R is defined as:

$$R = \frac{N_d}{N_d + N_o} \tag{A.10}$$

The CA is calculated by:

$$CA = \frac{N_d}{N_d + N_o + N_c} \tag{A.11}$$

For example, we have the following square confusion matrix M_c of n order (*cf.* equation A.12), whose coefficients m_{pq} represent the number of elements of class q assigned to cluster p.

$$M_c = \begin{pmatrix} m_{11} & m_{12} & m_{13} & \dots & m_{1i} & \dots & m_{1n} \\ m_{21} & m_{22} & m_{23} & \dots & m_{2i} & \dots & m_{2n} \\ m_{31} & m_{32} & m_{33} & \dots & m_{3i} & \dots & m_{3n} \\ \vdots & & & & & & \\ m_{i1} & m_{i2} & m_{i3} & \dots & m_{ii} & \dots & m_{in} \\ \vdots & & & & & & \\ m_{n1} & m_{n2} & m_{n3} & \dots & m_{ni} & \dots & m_{nn} \end{pmatrix} \tag{A.12}$$

Thus, the PT is considered as a weighted sum of individual cluster purities:

$$PT = \frac{\sum_{i=1}^{n}(PT_i \sum_{q=1}^{n} m_{iq})}{\sum_{p,q=1}^{n} m_{pq}} \tag{A.13}$$

where

$$PT_i = \frac{1}{\sum_{q=1}^{n} m_{iq}} \max_{1 \leq q \leq n} (m_{iq})$$

Likewise, the E is formulated as follows:

$$E = \frac{\sum_{i=1}^{n} (E_i \sum_{q=1}^{n} m_{iq})}{\sum_{p,q=1}^{n} m_{pq}} \tag{A.14}$$

where

$$E_i = -[\sum_{q=1}^{n} \frac{m_{iq}}{\sum_{q=1}^{n} m_{iq}} \log_2(\frac{m_{iq}}{\sum_{q=1}^{n} m_{iq}})]$$

The computation of the P and R is illustrated in Table A.2.

Table A.2.: Confusion Matrix.

Clustering outcomes \ Ground-truth	Class 1	Class 2	Class 3	...	Class i	...	Class n	
Cluster 1	m_{11}	m_{12}	m_{13}	...	m_{1i}	...	m_{1n}	$\leftrightarrow P_1$
Cluster 2	m_{21}	m_{22}	m_{23}		m_{2i}		m_{2n}	$\leftrightarrow P_2$
Cluster 3	m_{31}	m_{32}	m_{33}		m_{3i}		m_{3n}	$\leftrightarrow P_3$
...								$\leftrightarrow$ ...
Cluster i	m_{i1}	m_{i2}	m_{i3}		m_{ii}		m_{in}	$\leftrightarrow P_i$
...								$\leftrightarrow$ ...
Cluster n	m_{n1}	m_{n2}	m_{n3}	...	m_{ni}	...	m_{nn}	$\leftrightarrow P_n$
	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	
	R_1	R_2	R_3	...	R_i	...	R_n	

Since the used pixel-based clustering technique is unsupervised, the cluster label attributed by the clustering technique may be different from our specified ground-truth. Thus, we manage the correspondence between the cluster label and ground-truth to compute the M_c and calculate afterwards the different classification accuracy metrics. If the lines of the confusion matrix are switched or interchanged that the cluster i corresponds to the class j, it is then possible to define P_i and R_j, the precision of the cluster i and the recall of the class j, respectively. For a class "Class j", the individual cluster precision (P_i) assesses the rate of pixels assigned and classified as "Class j" which do not belong to "Class j" as defined in the ground-truth. On the other hand, the individual cluster recall (R_j) evaluates the percentage of the pixels, labeled "Cluster i" in the ground-truth which have been omitted by using the proposed pixel-labeling scheme for comparing texture features, *i.e.* they have not been classified as "Class j".

Therefore, the P metric corresponds to the proportion of the predicted cases that are correctly matched to the benchmark classifications. It is considered as a means of assessing the classification. The P is given by:

$$P = \frac{1}{n} \sum_{i=1}^{n} P_i = \frac{1}{n} \sum_{i=1}^{n} P(i,j) = \frac{1}{n} \sum_{i=1}^{n} \frac{m_{ij}}{\sum_{q=1}^{n} m_{iq}} \tag{A.15}$$

The R measure indicates the proportion of real cases that are correctly predicted. It is considered a way to improve the classification. The R is given by:

$$R = \frac{1}{n} \sum_{j=1}^{n} R_j = \frac{1}{n} \sum_{j=1}^{n} R(i,j) = \frac{1}{n} \sum_{j=1}^{n} \frac{m_{ij}}{\sum_{p=1}^{n} m_{pj}} \tag{A.16}$$

The CA metric corresponds to the ratio of the true classified predicted pixels and the total number

of pixels. The CA is given by:

$$CA = \frac{\sum_{p=1}^{n} m_{pp}}{\sum_{p,q=1}^{n} m_{pq}} \tag{A.17}$$

Finally, the F-measure (F) can be computed as a score resulting from the combination of the P and R accuracies by using a harmonic mean. It assesses both the homogeneity and the completeness criteria of a clustering result. The F is given by:

$$F = \frac{2\ P\ R}{P + R} \tag{A.18}$$

The different clustering and classification accuracy metrics proposed in the literature for performance evaluation are summarized in Table A.3.

Table A.3.: Clustering and classification accuracy metrics in the literature.

Index	Description
A- Internal or unsupervised clustering accuracy metrics	
Compactness [535]	This measure is used to evaluate the variance of the proximity matrix of clusters samples.
Separation [535]	This index quantifies the distance between two different clusters using the three following approaches: distance between the closest member of the clusters, distance between the most distant members and distance between the clusters centers.
Silhouette width index [341]	This index measures the level of compactness and separation by analyzing the distribution of the observations into clusters.
Dunn index [397]	This index is used to compute the ratio of the minimal inter-cluster distance to the maximal intra-cluster one in order to determine the dense and well-separated clusters.
Davies-Bouldin index [391]	This metric is used to calculate the ratio of the sum of within-cluster scatter to between-cluster separation.
Homogeneity [552]	This metric uses the average similarity between cluster members with respect to clustering.
Separation [552]	This index defines the separation of a clustering as the average dissimilarity between two clusters.
Cubic Clustering Criterion [368]	This metric quantifies the deviation of the clusters from the distribution by computing the within-cluster of the sum-of-squares and cross-products matrix of data.
Krzanowski-Lai index [382]	This index evaluates the between- and within-cluster sums of squares of the partition.
Hartigan index [383]	This metric is performed using the within-group dispersion matrix for data clustered into k clusters.
Calinski-Harabasz index [384]	This accuracy is performed based on the within-group dispersion of clusters by computing the cluster centers.
Scott index [385]	This index is evaluated by performing the ratio of the sum-of-squares within the clusters to the sum-of-squares and cross-products matrix of data.
Marriot index [386]	This metric evaluates the within-cluster sums-of-squares of the partition by examining the effect of adding a single point into a data set.
TraceCovW index [387]	This index is computed using the covariance of the within-cluster of the sum-of-squares and cross-products matrix of data.
TraceW index [387]	This index is computed by evaluating the within-cluster of the sum-of-squares and cross-products matrix of data.
Friedman index [388]	This index which is known as the index of coincidence, is used to explore the structure of heterogeneous multivariate data based on the non-singular linear transformations.
Rubin index [389]	This metric is based on the computation of the within-cluster matrix.
C-index [390]	This index is based on the within-cluster dissimilarities.
Ratkowsky index [392]	This accuracy is quantified by computing the sum-of-squares between the clusters.
Ball index [393]	This index is computed based on the average distance between cluster members and their respective cluster centroids.
PtBiserial index [394]	This index is performed using the within- and between- cluster distances.
Frey index [395]	This index is based on the computation of the ratio of the difference scores from two successive levels in the hierarchy of the applied hierarchical algorithm.
McClain index [396]	This metric considers the ratio of the average within-cluster distance by the number of cluster distances.
SDindex [398]	This index explores the concepts of the average scattering for clusters and separation between clusters.
SDbw validity index [399]	This accuracy defines the criteria of compactness and separation between clusters.
Weighted inter-intra measure [553]	This accuracy is used to compare the homogeneity of data to its separation.

Table A.3 – continued from previous page

Index	Description
Huberts Γ statistic [554]	The aim of this metric is to find the degree of match between a given clustering scheme and its proximity matrix.
B- External or supervised clustering accuracy metrics	
Rand index [401]	This index quantifies the similarity between the distributions of the observations in the clustering result and the benchmark classifications.
Jaccard coefficient [342]	This coefficient is used to assess the similarity between the distributions of the observations in the clustering result and in the ground-truth.
Fowlkes-Mallows index [405]	This metric compares the distributions of the observations in the clustering result and in the ground-truth by measuring the probability that a pair of observations is classified together in both the clustering solution and the ground-truth class.
Adjusted Rand index [402]	This measure is a similar form of the Rand index which is adjusted in order to attenuate the role of the chance grouping of cluster members by providing a correction for chance agreement.
Mutual information measure [403]	This metric quantifies the shared information between the distribution of a clustering and a ground-truth classification.
Adjusted mutual information measure [404]	This measure is considered as a variation of mutual information measure which corrects the effect of agreement of cluster members.
Mirkin metric [555]	This metric performs afterward the Hamming distance between the distributions of observations in the clustering result and in the ground-truth.
V-measure [556]	This measure is an external entropy based cluster evaluation accuracy which explicitly quantifies the degree of satisfaction of homogeneity and completeness criteria.
C- Classification accuracy metrics related to the confusion matrix	
Precision [347]	This accuracy corresponds to the proportion of predicted cases that are correctly matched to the benchmark classifications. It is considered as a means of assessing the classification.
Recall [347]	This measure indicates the proportion of the real cases that are correctly predicted. It is considered a way to improve the classification.
F-measure [347, 557]	This measure is considered as a score resulting from the combination of the precision and recall accuracies by using a harmonic mean. It assesses both the homogeneity and the completeness criteria of a clustering result.
Classification accuracy [558]	This measure denotes the quotient of true classified cluster members and the total number of data observations.
Entropy [559]	This index evaluates the confusion level through the confusion matrix from the class distribution of misclassified pixels.
Purity [559]	The purity of a cluster corresponds to the ratio of the largest class of pixels assigned to this cluster with respect to the overall cluster size.

A.3. Clustering evaluation or validity indices for the estimation of the number of clusters in the literature

Table A.4.: Clustering evaluation or validity indices for the estimation of the number of clusters in the literature.

Index	**Optimal number of clusters**	**Description**
Silhouette width index [341]	Maximum value of the index	This index measures the level of compactness and separation by analyzing the distribution of the observations into clusters.
Dunn index [397]	Maximum value of the index	This index computes the ratio of the minimal inter-cluster distance to the maximal intra-cluster one in order to determine the dense and well-separated clusters.
Davies-Bouldin index [391]	Minimum value of the index	This metric calculates the ratio of the sum of within-cluster scatter to between-cluster separation.
Cubic Clustering Criterion [368]	Maximum value of the index	This metric quantifies the deviation of the clusters from the distribution by computing the within-cluster of the sum-of-squares and cross-products matrix of data.
Krzanowski-Lai index [382]	Maximum value of the index	This index evaluates the between- and within-cluster sums of squares of the partition.
Hartigan index [383]	Maximum difference between hierarchy levels of the index	This metric is performed by using the within-group dispersion matrix for data clustered into k clusters.
Calinski-Harabasz index [384]	Maximum value of the index	This accuracy is performed based on the within-group dispersion of clusters by computing the cluster centers.
Scott index [385]	Maximum difference between hierarchy levels of the index	This index is evaluated by performing the ratio of the sum-of-squares within the clusters to the sum-of-squares and cross-products matrix of data.
Marriot index [386]	Maximum value of second differences between levels of the index	This metric evaluates the within-cluster sums-of-squares of the partition by examining the effect of adding a single point into a data set.
TraceCovW index [387]	Maximum difference between hierarchy levels of the index	This index is computed by using the covariance of the within-cluster of the sum-of-squares and cross-products matrix of data.
TraceW index [387]	Maximum value of absolute second differences between levels of the index	This index is computed by evaluating the within-cluster of the sum-of-squares and cross-products matrix of data.
Friedman index [388]	Maximum difference between hierarchy levels of the index	This index which is known as the index of coincidence, explores the structure of heterogeneous multivariate data based on the non-singular linear transformations.
Rubin index [389]	Minimum value of second differences between levels of the index	This metric is based on the computation of the within-cluster matrix.
C-index [390]	Minimum value of the index	This index is based on the within-cluster dissimilarities.
Ratkowsky index [392]	Maximum value of the index	This accuracy is quantified by computing the sum-of-squares between the clusters.
Ball index [393]	Maximum difference between hierarchy levels of the index	This index is computed based on the average distance between cluster members and their respective cluster centroids.
PtBiserial index [394]	Maximum value of the index	This index is performed by using the within- and between- cluster distances.
Frey index [395]	The cluster level before that index value < 1	This index is based on the computation of the ratio of the difference scores from two successive levels in the hierarchy of the applied hierarchical algorithm.
McClain index [396]	Minimum value of the index	This metric considers the ratio of the average within-cluster distance by the number of cluster distances.
SDindex [398]	Minimum value of the index	This index explores the concepts of the average scattering for clusters and separation between clusters.
SDbw validity index [399]	Minimum value of the index	This accuracy defines the criteria of compactness and separation between clusters.

Appendix B.

Detailed description of some parts of the work presented in this dissertation

Contents

B.1. A summary of the analyzed texture features in this work

This section presents an exhaustive and detailed review of the different analyzed texture features which have been carried in this work. First, for each set of texture descriptors a state-of-the-art related to the parametrization of the used texture features in the most explored fields in image analysis and pattern recognition, with a particular focus on those related to sub-fields and tasks of DIA and historical DIA, is briefly presented. Then, a detailed review of the texture features and their parameters is discussed. Finally, we conclude by detailing and justifying the techniques and parameters used in our study based on work published in the literature and after performing several experiments to choose the best configuration of the pre-defined thresholds and parameters

B.1.1. Tamura features

The first set of texture features investigated in this work is the Tamura descriptors.

B.1.1.1. Generalities and related works

Tamura *et al.* [159] proposed to extract textural features corresponding to human visual perception. They defined six basic texture descriptors, namely coarseness, contrast, directionality, line-likeness, regularity and roughness. They proved that the three first textural features (*i.e.* coarseness, contrast and directionality) consistently outperformed others for global descriptions of textures both separately and in combinations for image segmentation and classification issues.

The Tamura features have mainly been used in computer vision applications, such as content-based image retrieval [560, 561, 562]. Paulhac *et al.* [563] used the texture attributes as statistical measures for characterizing the image contrast. The texture descriptors were extracted from several resolutions, based on the Tamura descriptors for real 3-D ultra-sound images. Zhang *et al.* [564] showed that the Tamura features are efficient and robust to locate the license plates after retrieving the candidate horizontal regions of license plate by applying the run-length technique. Recently, the Tamura descriptors have been extracted to assist DIA. Keysers *et al.* [214] compared several texture features, including the Tamura texture features histogram, relational invariant feature histogram, run-length histogram, distribution of connected components, *etc.* for DI zone classification. They concluded that the Tamura features are the single best ones but they have high demand in computational time (*i.e.* more than 100 times slower to compute than the most other extracted descriptors). Mouats *et al.* [258, 259] introduced the Tamura descriptors in their Gabor-based segmentation of HDIs method to improve the obtained results.

B.1.1.2. Tamura Features

Four Tamura descriptors are extracted in this work, namely:

- Coarseness (*cf.* equation B.4),
- Contrast (*cf.* equation B.5),
- Number of orientations (*cf.* equation B.11),
- Directionality (*cf.* equation B.12).

The following details the four extracted Tamura descriptors.

1. ***Coarseness***
   The coarseness feature is considered by Tamura *et al.* [159] as the most fundamental texture feature. It illustrates the scale and repetition rates of texture. Specifically, the coarseness feature measures the largest size at which a texture exists. For instance, when two images differ only in scale, the magnified one is coarser (*cf.* Figure B.1). Moreover, when two images

have different structures, the bigger its pattern size and/or less its patterns are repeated, the coarser texture (*cf.* Figure B.2). Measures of coarseness are presented at the bottom of each image in Figures B.1 and B.2.

Figure B.1.: Illustration of the texture coarseness on an example of a scaled drop cap. Figures (a),(b) and (c) are the original image, its reduced one ($S_I \div 2$) and its magnified one ($S_I \times 2$), respectively, where S_I is the size of the original image.

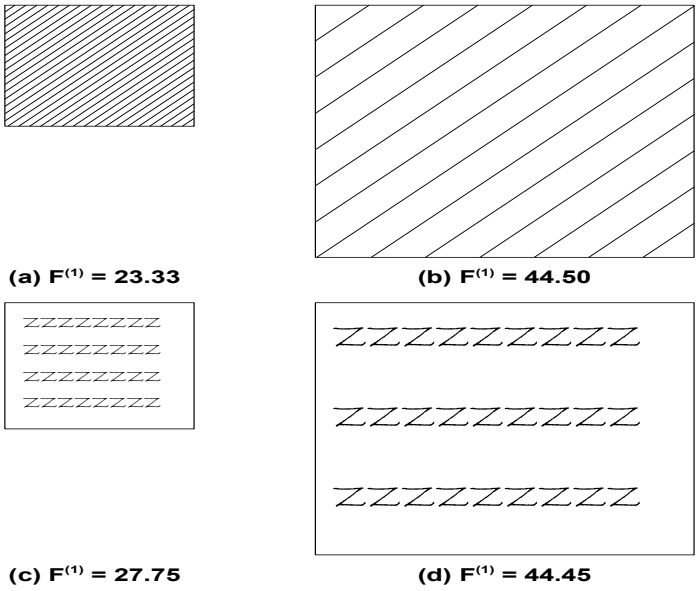


Figure B.2.: Illustration of the texture coarseness on two images having different structures. Figures {(a) and (c)} are the original images, and Figures {(b) and (d)} are their edited versions with large pattern size and less repeated pattern, respectively.

The coarseness is firstly computed by taking the average $A_{k_t}(x, y)$ at every image pixel $I(x, y)$ over the neighborhood of size $2^{k_t} \times 2^{k_t}$ according to the following equation:

$$A_{k_t}(x, y) = \sum_{i=x-2^{k_t-1}}^{x+2^{k_t-1}-1} \sum_{j=y-2^{k_t-1}}^{y+2^{k_t-1}-1} \frac{f(i, j)}{2^{2k_t}} \tag{B.1}$$

where $f(x, y)$ represents the gray-level of image pixel $I(x, y)$ and $k_t \in [1, L]$ where $2^L \leq \min(W, H)$, W and H denote the effective width and height of the analyzed image.

Secondly, at each pixel the differences $E_{k_t,h}(x,y)$ and $E_{k_t,v}(x,y)$ between the average of pairs corresponding to pairs of non-overlapping neighborhoods on opposite sides of the analyzed pixel in both the horizontal and vertical orientations, respectively, are computed as:

$$E_{k_t,h}(x,y) = |A_{k_t}(x + 2^{k_t-1}, y) - A_{k_t}(x - 2^{k_t-1}, y)| \tag{B.2}$$

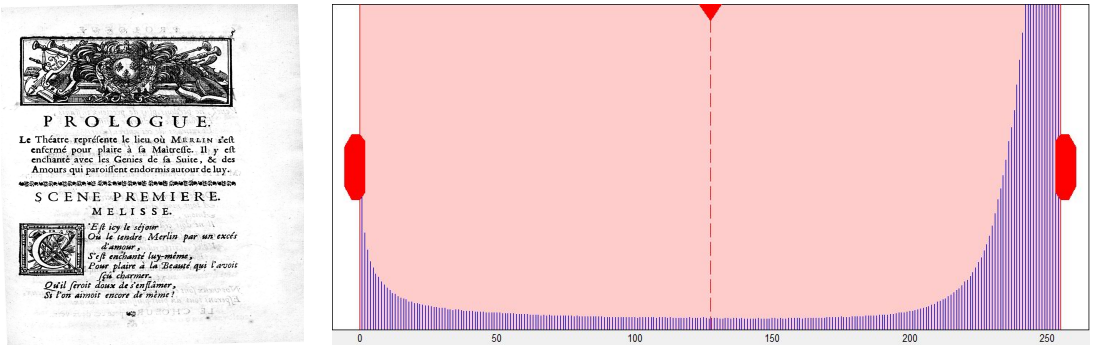
$$E_{k_t,v}(x,y) = |A_{k_t}(x, y + 2^{k_t-1}) - A_{k_t}(x, y - 2^{k_t-1})| \tag{B.3}$$

Thirdly, the best size $S_{best}(x,y) = 2^{k_t}$ is defined according to the specified k_t which maximized $E = E_{max} = max_{1 \leq k_t \leq L}(E_{k_t,h}(x,y), E_{k_t,v}(x,y))$ in either the horizontal direction or vertical one.

Finally, the coarseness measure is defined as the average of S_{best} over the analyzed image according to the equation B.4.

$$F^{(1)} = \frac{1}{WH} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} S_{best}(x,y) \tag{B.4}$$

2. ***Contrast***
The contrast feature measures the dynamic range of gray-levels in an image with taking into consideration the distribution polarization of black and white pixels (*i.e.* black-to-white mapping) (*cf.* Figure B.3).



(a) Original image (b) Black-to-white mapping

Figure B.3.: Illustration of the black-to-white mapping to estimate the dynamic range of gray-levels for contrast adjustment. Figure (a) is the original image. Figure (b) represents the black-to-white mapping to estimate the dynamic range of gray-levels for contrast adjustment.

The contrast is given by the equation B.5.

$$F^{(2)} = \frac{\sigma^2}{(\mu_4)^{\frac{1}{4}}} \tag{B.5}$$

where μ_4 is the fourth moment and σ represents the standard deviation estimator.

3. ***Number of orientations***
By building the histogram of local edge probabilities $Hist_D$, global texture features such as long lines and simple curves can be characterized (*cf.* Figure B.4).

Two 3×3 masks are firstly applied horizontally ∇_H (*cf.* equation B.6) and vertically ∇_V (*cf.* equation B.7):

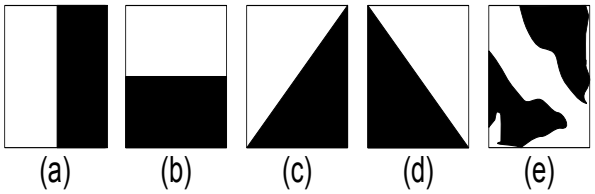


Figure B.4.: Illustration of few edge kinds for building the histogram of local edge probabilities. Figures (a), (b), (c), (d) and (e) represent vertical, horizontal, 45°, 135° and non-directional edges.

$$\nabla_H = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \qquad \text{(B.6)} \qquad\qquad \nabla_V = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \qquad \text{(B.7)}$$

Then, image edges can be detected by extracting magnitude $|\Delta G|$ (*cf.* equation B.8) and direction θ_t (*cf.* equation B.9) at each pixel.

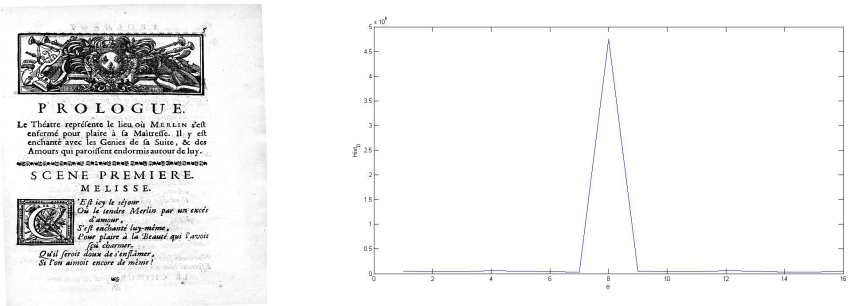
$$|\Delta G| = \frac{|\nabla_V| + |\nabla_H|}{2} \qquad \text{(B.8)}$$

$$\theta_t = \tan^{-1} \frac{\nabla_V}{\nabla_H} + \frac{\pi}{2} \qquad \text{(B.9)}$$

Therefore, $Hist_D$ is produced by quantifying θ_t and counting all pixels respecting the following condition: $|\Delta G| \geq t_{Hist}$. t_{Hist} and n_b are the specified $Hist_D$ threshold and the number of the $Hist_D$ bins which are set to 12 and 16, respectively (*cf.* Figure B.5). $Hist_D$ is defined to be:

$$Hist_D(l) = \frac{N_{\theta_t}(l)}{\sum_{i=0}^{n_b - 1} N_{\theta_t}(i)} \qquad \text{(B.10)}$$

where $l = 0, 1, \ldots, n_b - 1$. $N_{\theta_t}(l)$ is the number of pixels at which $\frac{(2l-1)\Pi}{2n_b} \leq \theta_t < \frac{(2l+1)}{2n_b}$.



(a) Original image (b) Histogram of local edge probabilities

Figure B.5.: Illustration of the histogram of local edge probabilities. Figure (a) is the original image. Figure (b) represents the histogram of local edge probabilities corresponding to the original image.

Therefore, the number of orientations describes the local edge density and distribution which is given by extracting salient histogram peaks (*i.e.* local histogram maxima) after computing the difference vector between two successive histogram bins, according to the equation B.11.

$$F^{(3)} = \sum_{k} [\underset{0 \leq k \leq n_b - 1}{\operatorname{argmax}} (\frac{\partial Hist_D(k)}{\partial k} = 0)] \tag{B.11}$$

4. ***Directionality***
   The directionality feature provides an insight into the global texture property over a region by measuring the total degree of texture directionality. It is computed by using a histogram of local edge probabilities $Hist_D$ against their directional angle (*cf.* Figure B.6).

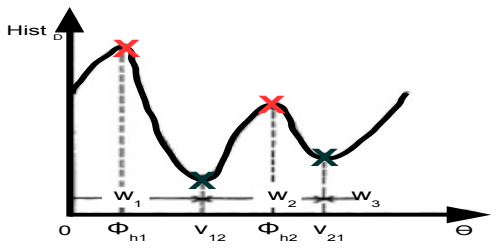


Figure B.6.: Illustration of the computation of the directionality feature from the histogram of local edge probabilities.

By quantifying the sharpness of $Hist_D$ peaks, the texture directionality is measured by summing the second moments around each peak according to the equation B.12 .

$$F^{(4)} = 1 - r\ n_p \sum_{p}^{n_p} \sum_{\Phi_h \in w_p} (\Phi_h - \Phi_p)^2 Hist_D(\Phi_h) \tag{B.12}$$

where n_p, Φ_p, w_p, r and Φ_h represent the number of histogram peaks which was set by Tamura *et al.* [159] to 2, the p^{th} peak position of $Hist_D$, the range of p^{th} peak between valleys, the normalizing factor related to the quantized levels of Φ_h and the quantized direction code (cyclically in modulo 180°), respectively.

B.1.2. LBP features

The second set of texture features investigated in this work is the LBP descriptors.

B.1.2.1. Generalities

The LBP descriptors are extracted from the LBP operator. The LBP operator is one of the most explored local image descriptor for texture analysis which has mainly used for describing local texture properties of gray-scale images. It has been introduced to measure pure and original property of the texture spectrum by Wang and He [260]. They proposed a texture analysis pattern based on a texture unit. LBP is a two-level version of the texture spectrum method. Later, it was popularized by Ojala *et al.* [261] and Harwood *et al.* [262] to analyze texture characteristics for texture classification. Ojala and Pietikäinen [263] presented an unsupervised texture segmentation method based on examining the LBP distributions.

LBP is obtained by locally thresholding texture and their combinations with local gray-scale measures. It represents each analyzed image pixel with a binary pattern based on the difference between its gray-level value and its circular neighborhood with specified radius R_l. If the gray-level value difference between the analyzed pixel $I_c(x, y)$ and its P_l neighboring pixels $I_{p \in [0, P_l - 1]}(x, y)$, is greater than or equal to zero, the LBP value is set to 1, otherwise is set to 0 (*cf.* Figure B.7(a)). Thus, the resistance to the intensity value of pixels in gray-scale format is ensured.

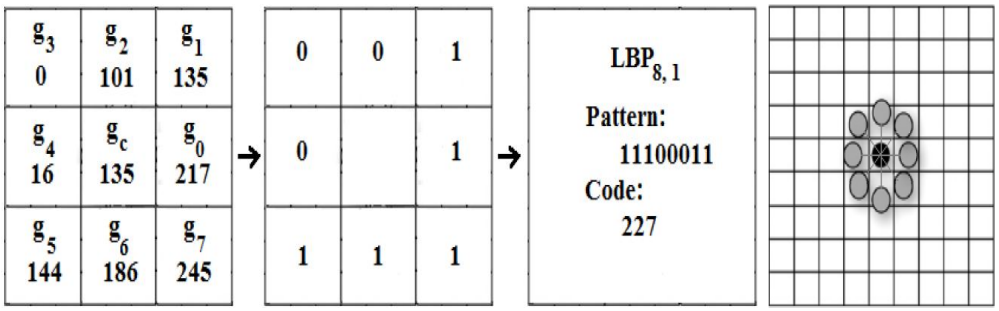
If the coordinates of the analyzed pixel are $(0,0)$, then the coordinates of $I_p(x,y)$ are given by $(-R_l \sin(\frac{2\Pi p}{P_l}), R_l \cos(\frac{2\Pi p}{P_l}))$. The interpolation is applied when the gray-level values of neighbors mismatches to an image pixel integer value. Then, by multiplying the binary elements with a binomial coefficient, the LBP value $0 \leq LBP_{P_l,R_l}(I_c(x,y)) \leq 2^{P_l}$ is produced which corresponds to the value of the LBP feature vector. The LBP operator LBP_{P_l,R_l} is defined according to the following equation:

$$LBP_{P_l,R_l}(I_c(x,y)) = \sum_{p=0}^{P_l-1} s(f_p(x,y) - f_c(x,y))2^p \tag{B.13}$$

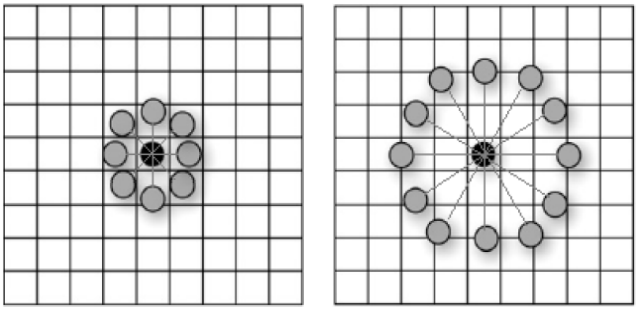
where

$$s(z) = \begin{cases} 1, \text{ if } z \geq 0, \\ 0, z < 0 \text{ otherwise.} \end{cases} \tag{B.14}$$

where P_l is the number of neighboring pixels in a circular set. $f_{p\in[0,P_l-1]}(x,y)$ corresponds to the gray-level values of equally spaced pixels from $I_c(x,y)$ on a circle of radius R_l which builds the P_l circularly symmetric neighbors $I_{p\in[0,P_l-1]}(x,y)$. $f_c(x,y)$ and $f_p(x,y)$ represent the gray-levels of the analyzed image pixel $I_c(x,y)$ and image pixel $I_p(x,y)$, respectively (*cf.* Figure B.7).



(a) Computation of the $LBP_{P_l=8,R_l=1}$ code



(b) $LBP_{P_l=12,R_l=2.5}$ (c) $LBP_{P_l=16,R_l=4}$

Figure B.7.: Illustration of the process of calculating the LBP operator LBP_{P_l,R_l}. Figure (a) shows an example of the computation of the $LBP_{P_l=8,R_l=1}$ code. Figures (b) and (c) illustrate two different circularly symmetric gray-level neighborhood sets around a central black pixel for different ($LBP_{P_l=12,R_l=2.5}$ and $LBP_{P_l=16,R_l=4}$) [234].

By taking into account P_l pixels in the neighbor set when computing a basic LBP_{P_l,R_l} operator, 2^{P_l} different binary patterns are obtained. The obtained 2^{P_l} binary patterns are not rotationally invariant. Thereby, by performing a circular bit-wise right-shift on the p-bit binary pattern and selecting the minimum value of $P_l - 1$ bit-wise right-shift operations on the binary pattern (*i.e.* assigning a unique identifier to each rotation invariant LBP), n_l unique rotation invariant LBP are produced to remove the effect of rotation. Indeed, the quantification of the occurrence statistics of the individual rotation invariant patterns corresponding to the image micro-features is ensured.

The rotation invariant LBP operator $LBP^{ri}_{P_l,R_l}$ is defined according to the following equation:

$$LBP^{ri}_{P_l,R_l}(I_c(x,y)) = \min_{0\leq i\leq P_l-1}\{ROR(LBP_{P_l,R_l}(I_c(x,y),i))\} \tag{B.15}$$

where $ROR(.,i)$ represents a circular bit-wise right-shift on the P_l-bit binary pattern i times.

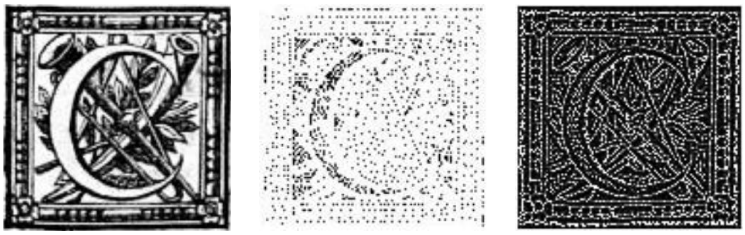
Noting that the obtained LBP feature vector is non-uniform, Ojala *et al.* [153] proposed an efficient multi-scale approach based on uniform LBP for gray-scale and rotation invariant texture classification. They proved that the basic 3×3 LBP operator provides better performance by extracting the uniform and non-uniform patterns from it. A pattern is considered as a uniform, if the number of spatial transitions (bit-wise 0/1 changes) in the pattern are less than or equal to 2. Therefore, the rotation invariant uniform 2 LBP operator is labeled "riu2". Formally, the rotation invariant uniform 2 LBP operator $LBP^{riu2}_{P_l,R_l}$ is defined according to the following equation:

$$LBP^{riu2}_{P_l,R_l}(I_c(x,y)) = \begin{cases} \sum_{p=0}^{P_l-1} s(g_p - g_c), \text{ if } U(LBP_{P_l,R_l}(I_c(x,y))) \leq 2, \\ \\ P_l + 1, \text{ otherwise.} \end{cases} \tag{B.16}$$

where

$$U(LBP_{P_l,R_l}(I_c(x,y))) = |s(g_{P_l-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P_l-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \tag{B.17}$$

Figures B.8(b) and B.8(c) illustrate the results of the application of the $LBP_{P_l=8,R_l=1}$ and $LBP^{riu2}_{P_l=8,R_l=1}$ operators on a drop cap, respectively. We can note that the $LBP^{riu2}_{P_l=8,R_l=1}$ operator provides better performance than the $LBP_{P_l=8,R_l=1}$ one (*i.e.* visually, there is a discernible difference to the naked eye with the two output images since we can see the different shapes and patterns in the output image of the application of the $LBP^{riu2}_{P_l=8,R_l=1}$ operator).



(a) Input image (b) Output image of $LBP_{P_l=8,R_l=1}$ (c) Output image of $LBP^{riu2}_{P_l=8,R_l=1}$

Figure B.8.: Illustration of the application of the $LBP_{P_l=8,R_l=1}$ and $LBP^{riu2}_{P_l=8,R_l=1}$ operators on a drop cap image.

Thus, by using the rotation invariant uniform 2 LBP operator ($LBP^{riu2}_{P_l,R_l}$), $P_l + 1$ uniform binary patterns are produced in a circularly symmetric neighbor set of P_l pixels. Each uniform binary pattern are labeled differently (*i.e.* a unique label is assigned to each uniform binary pattern corresponding to the number of "1" bits in the pattern ($0 \rightarrow P_l$)), while the non-uniform patterns are grouped in the "miscellaneous" label P_l+1. Hence, by using the $LBP^{riu2}_{P_l,R_l}$ operator as gray-scale invariant measure of texture characteristics of an image, the distribution of the binary patterns for the whole analyzed image is described by computing the histogram of binary patterns $Hist_{P_l,R_l}$. Ojala *et al.* [153] proved that the uniform patterns are frequently dominant in the distribution of the binary patterns compared to non-uniform ones. Therefore, non-uniform weights are assigned to the uniform and non-uniform patterns (*i.e.* a higher weight is assigned to the uniform patterns

and all the non-uniform patterns are grouped into single bin of $Hist_{P_l,R_l}$) which ensure better discrimination of spatial patterns. Thus, each uniform pattern is associated to a separate single $Hist_{P_l,R_l}$ bin while all the non-uniform patterns are assigned to another single $Hist_{P_l,R_l}$ bin.

For describing an image with $LBP^{riu2}_{P_l,R_l}$, a histogram of binary patterns $Hist_{P_l,R_l}$ of $P_l + 2$ bins is produced. Each bin provides an estimation of the probability to find the corresponding pattern in the analyzed image. For example, with $P_l = 8$ for each image pixel $I_c(x, y)$, $LBP_{8,R_l}(I_c(x, y))$, $LBP^{ri}_{8,R_l}(I_c(x, y))$ and $LBP^{riu2}_{8,R_l}(I_c(x, y))$ produce 256 unique binary patterns, 36 unique rotation invariant LBP and 10 $Hist_{P_l,R_l}$ bins, respectively (*cf.* Figure B.9). The number of the uniform and non-uniform patterns are 9 and 28, respectively.

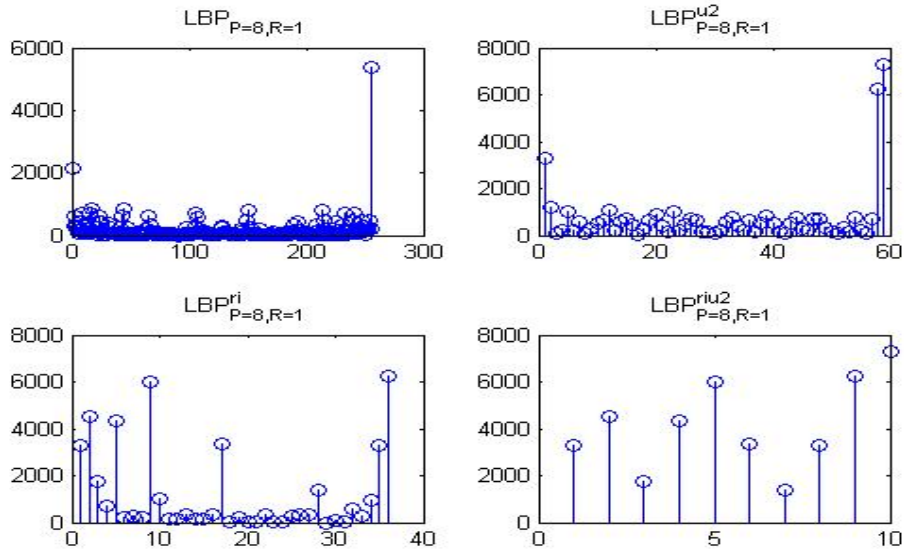


Figure B.9.: Representation of the drop cap image (*cf.* Figure B.8(b)) with the different histograms of binary patterns $Hist_{P_l=8,R_l=1}$ corresponding to $LBP_{P_l=8,R_l=1}$, $LBP^{u2}_{P_l=8,R_l=1}$, $LBP^{ri}_{P_l=8,R_l=1}$ and $LBP^{riu2}_{P_l=8,R_l=1}$.

B.1.2.2. State-of-the-art related to LBP parametrization

Low computational complexity, invariance to changes in the average intensity value of the central pixels comparing to its neighbors and ability to characterize fine texture details, make the LBP operator as one of the most widely used textural approach. LBP has been studied in a large variety of pattern recognition fields and has been successfully used by researchers in various contexts (e.g. medicine [565], face recognition [566], biometric [567]). More recently, the LBP operator has gained great attention of many researchers in the DIA fields. Dua *et al.* [264] extracted the LBP wavelet domain for off-line and text-independent writer identification. Lutf *et al.* [265] proposed a LBP-based approach for writer identification using off-line Arabic handwriting. They computed the LBP histogram to extract handwriting features for each diacritic after retrieving all diacritics from the input image. Ferrer *et al.* [266] proposed an algorithm based on the LBP orientation for printed script identification. Since Nicolaou *et al.* [204, 205] worked on binary images as inputs, they presented an approach based on appropriate redundant oriented binary LBP operator for Arabic font recognition. Bhowmik and Kar [234] compared the rotation invariant uniform LBP operator with the variance measure for segmentation of historical machine printed DIs. They concluded that the LBP operator outperforms the variance measure for separating graphic regions from text ones.

Jiang *et al.* [206] used the $LBP_{P_l=8,R_l=1}$ operator for printer identification. They generated 59-dimensional histogram (a feature vector composed of 58 uniform patterns and 1 single non-

uniform pattern) from the LBP operator for each analyzed gray-scale pixel of a DI. Bertolini *et al.* [203] extracted the LBP features from the $LBP^{u2}_{P_l=8,R_l=2}$ operator for writer identification and verification. They proved that the used LBP operator which produces a feature vector of 59 components for each analyzed pixel, is fast and accurate. Nicolaou *et al.* [204, 205] introduced a redundant oriented LBP ($P_l = 8, R_l = 3$) for Arabic font recognition. They extracted 327 redundant LBP features, including 255 bins from the LBP histogram, 36 rotation invariant features, 8 rotation phase features, 14 edge features, 5 beta-function features and 9 sample-count features. Bhowmik and Kar [234] localized text in HDIs by extracting LBP_{P_l,R_l}, $LBP^{ri}_{P_l,R_l}$ and $LBP^{riu2}_{P_l,R_l}$ features. They used three LBP operators by setting R_l equal to 1, 2 and 3 and P_l equal to 8, 16 and 24, respectively. But, they considered only P_l equal to 8 during the binary pattern computation. They concluded that the LBP_{P_l,R_l} model outperforms slightly the two other models $LBP^{ri}_{P_l,R_l}$ and $LBP^{riu2}_{P_l,R_l}$. But, in the most cases, the obtained results of the three models are relatively similar.

B.1.2.3. LBP features

In this work, $LBP^{riu2}_{P_l=8,R_l=1}$ is applied and 10 $Hist_{P_l,R_l}$ bins is produced for each analyzed pixel to ensure better discrimination of spatial patterns. Indeed, 10 $LBP^{riu2}_{P_l=8,R_l=1}$ descriptors are extracted. The $LBP^{riu2}_{P_l=8,R_l=1}$ feature vector consists of 10 terms of the probability to find the corresponding pattern in the analyzed image. The nine first descriptors correspond to the nine $Hist_{P_l=8,R_l=1}$ bins which represent the uniform patterns (*cf.* equation B.18), while the last one represent the last $Hist_{P_l=8,R_l=1}$ bin which characterizes all the non-uniform patterns (*cf.* equation B.19).

1. ***Heights of the uniform bins of the histogram of binary patterns***
   The first nine LBP descriptors correspond to the uniform bins which characterize the uniform patterns of the analyzed DI region. They are obtained from the nine first $Hist_{P_l=8,R_l=1}$. Hence, the first nine LBP descriptors are defined by:

$$F^{(i=1\rightarrow P_l+1)} = Hist_{P_l,R_l}(i) \tag{B.18}$$

2. ***Height of the non-uniform bin of the histogram of binary patterns***
   The last LBP descriptor represents the last $Hist_{P_l=8,R_l=1}$ bin which characterizes all the non-uniform patterns. Hence, the last LBP descriptor is defined to be:

$$F^{(i=P_l+2)} = Hist_{P_l,R_l}(i = P_l + 2) \tag{B.19}$$

B.1.3. GLRLM features

The third set of texture features investigated in this work is the GLRLM descriptors.

B.1.3.1. Generalities

The GLRLM descriptors are extracted by applying the run-length method. The run-length method has been extensively studied in a wide array of fields for analysis of images and particularly for pattern recognition and texture classification [267]. It has been introduced by Galloway *et al.* [181] to classify a set of terrain samples by extracting various run-length features from several GLRLM.

For a given image, an element of the GLRLM $p(g, l)$ is defined as the number of runs with pixels of gray-level g and run-length l. A gray-level run g is a sequence in a scan direction of a set of consecutive and collinear image pixels with identical gray-level value. The length of the run l is the number of image pixels in the run. A GLRLM is computed for runs having any given direction. Usually, the four scan directions have been used: $\theta_r = \{0°, 45°, 90°, 135°\}$ (*i.e.* horizontal, vertical, diagonal and anti-diagonal directions). For the GLRLM, the dimension of g is equal to G^l which corresponds to the maximum gray-level (*i.e.* number of gray-level bins). On the other hand, the

dimension of l is equal to L which corresponds to the maximum run-length. An example of the process of calculating the GLRLM for runs having horizontal direction is presented in Figure B.10.

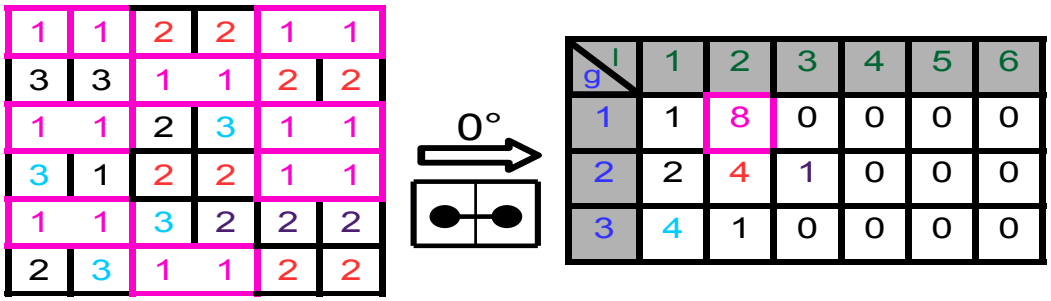


Figure B.10.: Illustration of the process of calculating the GLRLM for runs having horizontal direction (*i.e.* 0° direction).

In order to reduce the effect of noise and intensity fluctuations and overcome the problem of density representations, a step of quantization of gray-levels values is required. Thus, a gray-level run is considered as a contiguous sequence of image pixels defined in a scan direction, where pixel intensity gray-levels are defined in a certain range. For example, in the case of a gray-scale image which has 256 gray-levels, if a quantization of gray-levels values step in 16 gray-scale bins is introduced, the gray-level intensities of pixels will be ranged from 0 to 15, 16 to 31, 32 to 47, ..., 239 to 255. Afterwards, a 2-D run-length histogram ($Hist_{g,l}$) is produced for each scan direction, such one axis represented the run-length and the other axis illustrates the gray-level value or gray-level value bin (*cf.* Figure B.11). $Hist_{g,l}$ is a histogram of run-lengths. Therefore, since the $Hist_{g,l}$ is normalized, the probability of a specific run-length $P(g,l)$ can be defined according to the following equation:

$$\sum_{g=0}^{G^l-1}\sum_{l=1}^{L} P(g,l) = 1 \tag{B.20}$$

where G^l is the number of gray-level bins (*i.e.* number of bins into which the image has been quantized) g is the gray-level value bin, L is the maximum run-length, and l is the run-length.

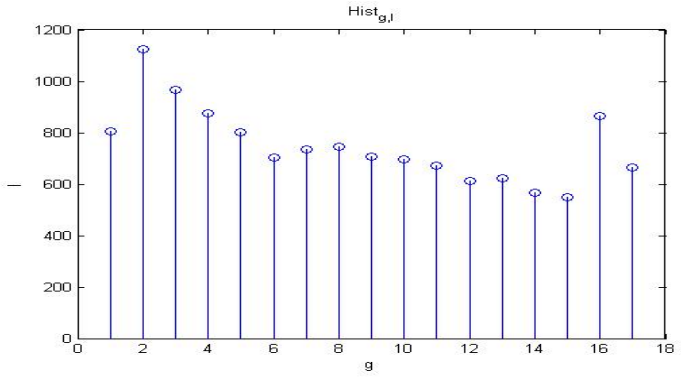


Figure B.11.: Illustration of the histogram of run-lengths $Hist_{g,l}$.

B.1.3.2. State-of-the-art related to GLRLM parametrization

The run-length method has been widely investigated in the analysis of biomedical images. For instance, Prasad and Sowmya [568] extracted textural features based on the GLRLM, GLCM, gray-level difference method and moments of gray-level histogram of a local area, for the analysis

of the human organs or tissues. Later, the run-length features were extracted for the recognition of the license plates [564]. The run-length technique was used to detect the horizontal regions of license plate. Although the poor performance of using the run-length or GLRLM features obtained by Weska *et al.* [250], and Conners and Harlow [268] comparing to classical texture features (GLCM, gray-level difference and the power spectrum features), the run-length methods have been recently applied to meet the need for DI segmentation or DIA, *etc.* Seuret *et al.* [223] proposed a method for discriminating printed content from handwritten annotations at pixel level. They extracted the run-length features in four directions $\theta_r = \{0°, 45°, 90°, 135°\}$ to estimate the width of a stroke in a given direction. Stamatopoulos *et al.* [269] used the run-length method for the page frame detection from double page DIs. They detected the vertical and horizontal zones of the two pages based on the vertical and horizontal white run projections, respectively. Nikolaou *et al.* [127] proposed an adaptive RLSA for the text line, word and character segmentation of historical and degraded machine-printed DIs. Although the proposed algorithm has been proved to work efficiently for a wide variety of degraded DIs, several thresholds were defined in the used segmentation techniques. Shi and Govindaraju [134] used a fuzzy run-length approach for the line separation in complex handwritten DIs including postal parcel images and historical handwritten DIs. Keysers *et al.* [214] proposed an accurate system for the classification of DIs based on the run-length feature extraction. The extracted features were used to classify text/non-text DI zones. Gordo *et al.* [215] used the multi-scale binarizing run-length histograms for the large-scale DI retrieval and classification. They worked on binary images as inputs, they quantized the lengths of the runs in logarithmic scale by defining 9 intervals for each quantized level (*i.e.* black and white gray-levels). Then, four run-length histograms were computed in horizontal, vertical, diagonal and anti-diagonal directions for each extracted region using spatial pyramids. The four run-length histograms were concatenated to characterize the extracted regions by a region descriptor of length $72 = 4$ directions $\times$ 2 quantized levels $\times$ 9 quantized intervals. The extracted descriptors have been proved that they work efficiently and do not require *a priori* knowledge of the DI layout, model, content or any kind of layout analysis. Dinstein and Shapira [270] extracted textural features based on the run-length histograms from groups of characters for the ancient Hebraic handwriting identification. The horizontal and vertical directions were selected to compute the run-length histograms. Then, the average dissimilarity between histograms of each writer was defined. Experiments yielded satisfying results. Another algorithm based on the run-length features was proposed for the handwriting identification on medieval DIs [271]. Uttama *et al.* [29] examined drop caps from historical heritage images and introduced a drop cap segmentation method based on a combination of different texture features extracted from the GLCM, GLRLM, auto-correlation function and Wold decomposition. Three run-length descriptors were extracted, including long-run emphasis (LRE), run percentage (RPC) and gray-level distribution.

B.1.3.3. GLRLM features

Galloway *et al.* [181] described a set of 11 texture features based on gray-level run-lengths and particularly the 2-D run-length histogram ($Hist_{g,l}$), to capture the coarseness of a texture in a specific direction:

- Short-run emphasis (SRE) (*cf.* equation B.21),
- Long-run emphasis (LRE) (*cf.* equation B.22),
- Low gray-level emphasis (LGRE) (*cf.* equation B.23),
- High gray-level emphasis (HGRE) (*cf.* equation B.24),
- Gray-level non-uniformity (GLNU) (*cf.* equation B.25),
- Run-length non-uniformity (RLNU) (*cf.* equation B.26),

- Run percentage (RPC) (*cf.* equation B.27),
- Short-run low gray-level emphasis (SRLGE) (*cf.* equation B.28),
- Long-run high gray-level emphasis (LRHGE) (*cf.* equation B.29),
- Short-run high gray-level emphasis (SRHGE) (*cf.* equation B.30),
- Long-run low gray-level emphasis (LRLGE) (*cf.* equation B.31).

In this work, for each analyzed foreground pixel four 2-D run-length histograms ($Hist_{g,l}$) are produced for each scan direction $\theta_r = \{0°, 45°, 90°, 135°\}$, *i.e.* horizontal, vertical, diagonal and anti-diagonal directions. For each $Hist_{g,l}$, a feature vector of 11 terms of GLRLM indices is computed.

1. ***Short-run emphasis (SRE)***
   SRE ensures the characterization of fine-grained textures by emphasizing short runs. SRE is defined by:
$$F^{(1)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} \frac{P(g,l)}{l^2} \tag{B.21}$$

2. ***Long-run emphasis (LRE)***
   LRE helps to characterize textures with large homogeneous areas or coarse textures by emphasizing long runs. LRE is defined by:
$$F^{(2)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} P(g,l) l^2 \tag{B.22}$$

3. ***Low gray-level emphasis (LGRE)***
   LGRE is orthogonal to SRE (*cf.* equation B.21) and it provides an insight of the dominance of many runs of low gray-level value in the analyzed texture. LGRE is defined by:
$$F^{(3)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} \frac{P(g,l)}{(g+1)^2} \tag{B.23}$$

4. ***High gray-level emphasis (HGRE)***
   HGRE is orthogonal to LRE (*cf.* equation B.22) and it provides information on the dominance of many runs of high gray-level value in the analyzed texture. HGRE is defined by:
$$F^{(4)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} P(g,l)(g+1)^2 \tag{B.24}$$

5. ***Gray-level non-uniformity (GLNU)***
   GLNU is focused on detecting the gray-level outliers from the histogram. GLNU is defined by:
$$F^{(5)} = \sum_{l=1}^{L} [\sum_{g=0}^{G^l-1} P(g,l)]^2 \tag{B.25}$$

6. ***Run-length non-uniformity (RLNU)***
   RLNU is an indicator of few run-length outliers which are dominating the histogram. RLNU

is defined by:

$$F^{(6)} = \sum_{g=0}^{G^l-1} [\sum_{l=1}^{L} P(g,l)]^2 \tag{B.26}$$

7. ***Run percentage (RPC)***
RPC gives a glimpse into the overall histogram homogeneity. The maximum RPC value corresponds to the case where all runs are equal to the unity length regardless of the gray-level values. RPC is defined by:

$$F^{(7)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} \frac{1}{P(g,l)l} \tag{B.27}$$

8. ***Short-run low gray-level emphasis (SRLGE)***
SRLGE is a combination of the two metrics: SRE (*cf.* equation B.21) and LGRE (*cf.* equation B.23) which estimates the dominance of many short runs of low gray-level value. SRLGE is defined by:

$$F^{(8)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} \frac{P(g,l)}{l^2(g+1)^2} \tag{B.28}$$

9. ***Long-run high gray-level emphasis (LRHGE)***
LRHGE is the complementary metric to SRLGE (*cf.* equation B.28). It characterizes the combination of long high gray-level value runs. LRHGE is defined by:

$$F^{(9)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} P(g,l)l^2(g+1)^2 \tag{B.29}$$

10. ***Short-run high gray-level emphasis (SRHGE)***
SRHGE is both orthogonal to SRLGE (*cf.* equation B.28) and LRHGE (*cf.* equation B.29). It carries out the domination of short runs with high intensity gray-levels in the analyzed texture. SRHGE is defined by:

$$F^{(10)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} \frac{P(g,l)(g+1)^2}{l^2} \tag{B.30}$$

11. ***Long-run low gray-level emphasis (LRLGE)***
LRLGE is the complementary metric to SRHGE (*cf.* equation B.30). It allows to characterize long runs with low intensity gray-levels in the analyzed texture. LRLGE is defined by:

$$F^{(11)} = \sum_{g=0}^{G^l-1} \sum_{l=1}^{L} \frac{P(g,l)l^2}{(g+1)^2} \tag{B.31}$$

B.1.4. Auto-correlation features

The fourth set of texture features investigated in this work is the auto-correlation descriptors.

B.1.4.1. Generalities

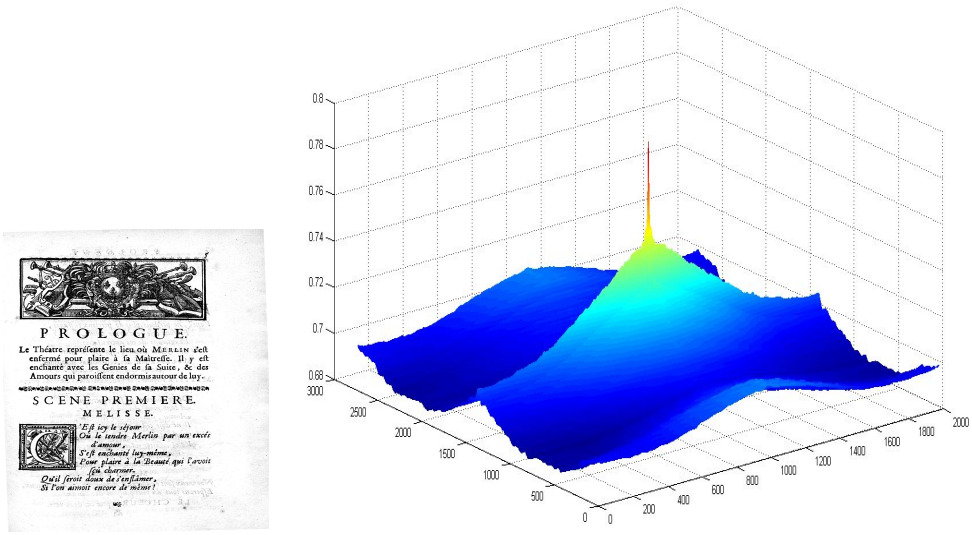
The auto-correlation features are extracted from a non-parametric tool which consists of the auto-correlation function. The auto-correlation function which is a 2-D function, is defined as a similarity

measure between a dataset and a shifted copy of the data. It is used to find periodic patterns and similar patterns through a number of extracted auto-correlation features [145, 179]. The auto-correlation function which is computed along the horizontal and vertical axes of the analysis window of an image I, is defined according to the following equation:

$$\begin{aligned} R_{(x,y)}^{I(\alpha,\beta)} &= \sum_{\alpha\in\Omega}\sum_{\beta\in\Omega} I(x,y)I(x+\alpha,y+\beta) \\ &= FFT^{-1}\left[FFT\left[I(x,y)\right]FFT^{*}\left[I(x,y)\right]\right] \end{aligned} \tag{B.32}$$

where $I(x+\alpha, y+\beta)$ is the translation of the analysis window of an image $I(x,y)$ by α and β pixels along the horizontal and vertical axes, respectively, defined on the plane Ω. FFT, $(.)^{*}$ and $(.)^{-1}$ denote the fast Fourier transform, complex conjugate and inverse transform, respectively.

An example of the application of the auto-correlation function on a HDI is presented on Figure B.12.



(a) Original image (b) Output image of the application of the auto-correlation function

Figure B.12.: Illustration of the application of the auto-correlation function on a HDI.

B.1.4.2. State-of-the-art related to auto-correlation parametrization

The auto-correlation function has extensively been investigated for texture analysis. For instance, Brown and Shvaytser [569] used the projective distortion of the auto-correlation function for determining local surface orientation. It has also been used by Heilbronner [570] for the analysis of the fabric and fine-grained materials, segmentation of the grain shapes and determination of the grain sizes. Lin *et al.* [571] characterized regular texture by computing periodicity based on extracting the auto-correlation primitives. By determining the location of peaks in the auto-correlation function applied on the gray-scale regular texture images, they determined if a texture image has or not a regular structure. Toyoda and Hasegawa [151] classified textures based on the extended higher order local auto-correlation features.

By analyzing the auto-correlation results, a rose of directions can be produced. The rose of directions which is a derivative of the auto-correlation function, is deduced from the auto-correlation function [273]. It is a polar diagram derived from the analysis of the auto-correlation results and reveals the significant orientations of the texture in the analyzed image block. It highlights interesting information concerning the principal orientations and periodicities of the texture, characterizing

the content of images without any assumption about page structure and its characteristics. The rose of directions has recently been used with HDIs [30, 1, 230]. In order to identify the main orientation of the analyzed image, the rose of directions is computed for each orientation by summing up the different values of the auto-correlation function (*cf.* equation B.32):

$$R^{I}_{(x,y)}(\Theta_i) = \sum_{D_i} R^{I(\alpha,\beta)}_{(x,y)} \tag{B.33}$$

where $\Theta_i \in [0, 180]$ is the selected orientation of the set of possible orientations D_i which is represented by a straight line passing through (x, y) and the angle Θ_i.

The rose of directions is normalized in one of the above studies in order to select only the relative variations of all contributions for each direction [1]. The relative sum $R'^{I}_{(x,y)}(\Theta_i)$ is defined as:

$$R'^{I}_{(x,y)}(\Theta_i) = \frac{R^{I}_{(x,y)}(\Theta_i) - R^{I}_{min}}{R^{I}_{max} - R^{I}_{min}} \tag{B.34}$$

where $R^{I}_{max} \neq R^{I}_{min}$, R^{I}_{min} and R^{I}_{max} represent the minimum and maximum values of $R^{I}_{(x,y)}(\Theta_i)$, respectively, both of which are computed on the analysis window of an image $I(x, y)$.

To illustrate the performance of the rose of directions in discriminating between textual and graphical regions in a DI and to determine the main orientation of a texture, Figure B.13 shows the rose of directions obtained with four different textures. As can be seen, the shape of the rose of directions is different for each type of texture. We note that for textual regions, the shape of the rose of directions depends on the orientation of the text and the main information. The horizontal orientation (0° and 180°) is clearly identifiable. On the other hand for drawing, the rose of directions is deformed.

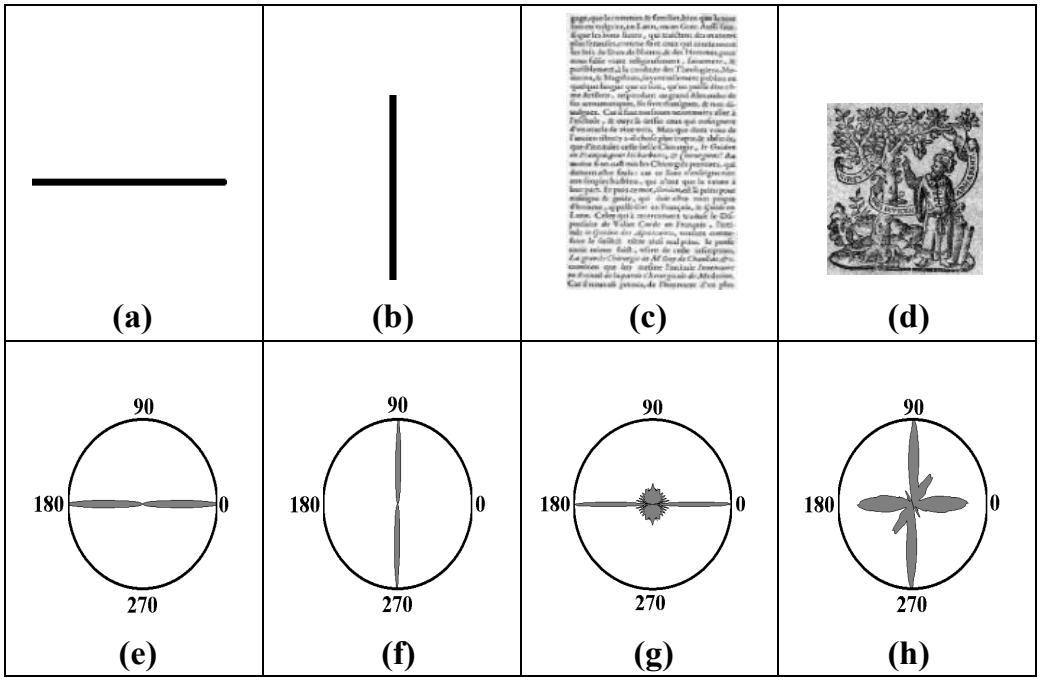


Figure B.13.: Examples of the rose of directions. Figures {(a), (b), (c) and (d)} are the original images, and Figures {(e), (f), (g) and (h)} represent their respective roses of directions. For textual regions such as in Figure (c), the shape of the rose depends on the orientation of the text and the main information. The horizontal orientation (0° and 180°) is clearly identifiable in Figure (g). For drawing in Figure (d), the rose of directions is deformed (*cf.* Figure (h)).

The various forms and shapes of the rose of directions which are obtained from the variety of textures contained in ancient gray-scale DIs do not help us to define a template of the rose of directions for each type of texture. Nevertheless, computing the rose helps us to extract significant and relevant indices for texture features. Journet *et al.* [1] defined texture features related to the orientation deduced from the rose of directions in order to analyze the digitized DI and to describe its content.

The use of the auto-correlation function is not new for the DIA community. Numerous studies have identified a number of auto-correlation features for segmenting HDIs and contemporary DIs [30, 1, 230, 245, 272, 229, 89]. Eglin *et al.* [30] determined the number of bank of GFs by selecting the relevant directions which were deduced from the rose of directions, to select interesting patterns for the noise reduction and classification of handwritings in ancient manuscripts. For historical DIA, Journet *et al.* [1] defined three auto-correlation features which few ones were derived from the rose of directions. The extracted features computed over the neighborhood of each pixel (foreground and background), were as follows: the main orientation of the rose of directions, the intensity value of the auto-correlation function for the main orientation and the variance in the intensities of the rose of directions, except for the main orientation. Grana *et al.* [245] used the auto-correlation matrix to distinguish between textual and pictorial regions in historical manuscripts. Garz and Sablatnig [230] presented a multi-scale texture-based approach for text region recognition in ancient manuscripts. They extracted the three auto-correlation features proposed firstly by Journet *et al.* [1] by applying three scales by means of overlapping sliding windows. Ouji *et al.* [272] introduced two other texture attributes (*i.e.* mean stroke width and height of an image), also in relation to the auto-correlation function for contemporary DI segmentation. For geometric layout analysis of HDIs, Coppi *et al.* [229] extracted the main regions from the page using the RXYC algorithm, then each region was divided in small squared blocks, and the local auto-correlation features were computed on each block and classified using a SVM classifier. The local auto-correlation features were deduced from a directional histogram obtained from the projections of the auto-correlation matrix along the vertical and horizontal directions in order to identify the repeating pattern of the texture. A 308-D feature vector for each block was constructed.

B.1.4.3. Auto-correlation features

The auto-correlation descriptors highlight interesting information on the principal orientations and periodicities of texture allowing characterizing the content of DIs without any assumption on the page layout, content, DI typographical or graphical characteristics. Thus, five auto-correlation features are extracted in this work [1, 272]:

- Main orientation of the rose of directions (*cf.* equation B.35),
- Intensity of the auto-correlation function for the main orientation (*cf.* equation B.36),
- Variance of the intensities of the rose of directions (*cf.* equation B.37),
- Mean stroke width along specific directions (*cf.* Algorithm 8),
- Mean stroke height along specific directions (*cf.* Algorithm 9).

The following details the five extracted auto-correlation descriptors.

1. ***Main angle of the rose of directions***
   The first texture feature $F^{(1)}_{(x,y)}$ corresponds to the main angle of the rose of directions extracted from its maximal intensity (Figure B.14). It is normalized by the deviation from the horizontal angle in order to avoid handling circular data. It is given by:

$$F^{(1)}_{(x,y)} = \left\| 180 - \underset{\Theta_i \in [0,180]}{\operatorname{argmax}} \left(R'^{I}_{(x,y)}(\Theta_i) \right) \right\| \tag{B.35}$$

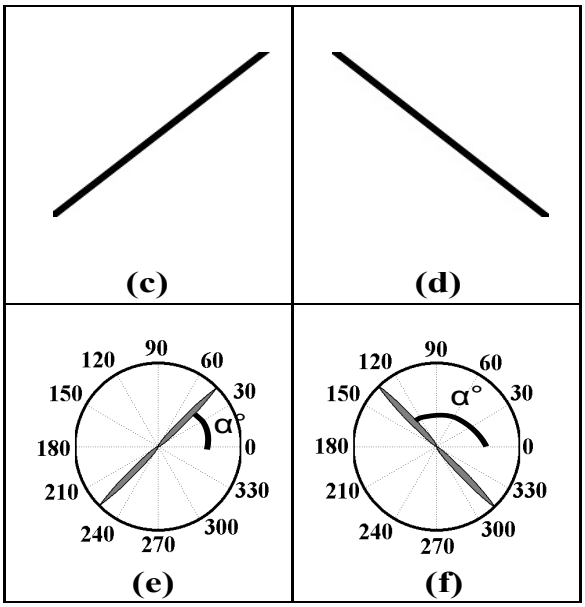


Figure B.14.: Examples of the main angle of the rose of directions extracted from its maximal intensity. {(c) and (d)} are the original images and {(e) and (f)} are their rose of directions, respectively. The main orientation on the rose of directions corresponds to the direction of the information contained in the analyzed image.

2. ***Intensity of the auto-correlation function for the main orientation***
   The second texture feature $F^{(2)}_{(x,y)}$ corresponds to the intensity of the auto-correlation function for the main orientation (*cf.* equation B.35) which is computed on the non-normalized value of the auto-correlation function (*cf.* equation B.33). This feature evaluates the anisotropy of an image $I(x, y)$ since the rose of directions associates the gray-level of pixels in a specific direction. It is computed as:

$$F^{(2)}_{(x,y)} = R^{I}_{(x,y)}(\underset{\Theta_i \in [0,180]}{\operatorname{argmax}}(R'^{I}_{(x,y)}(\Theta_i))) \tag{B.36}$$

3. ***Variance of the intensities of the rose of directions***
   The third texture index $F^{(3)}_{(x,y)}$ characterizes the overall shape of the rose of directions. $F^{(3)}_{(x,y)}$ is the variance of the rose intensities, except for the orientation of maximal intensity. A low $F^{(3)}_{(x,y)}$ means that the main orientation is significantly more prevalent than the other orientations. However, a high variance signifies that the rose is deformed and that there are a large number of orientations that are present to different extents (graphic blocks) (Figure B.15). Hence, the third texture descriptor is defined by:

$$F^{(3)}_{(x,y)} = \sigma^2(R'^{I}_{(x,y)}(\Theta_i)) \tag{B.37}$$

   where $\Theta_i \in [0, 180] \setminus \{\operatorname{argmax}_{\Theta_i \in [0,180]}(R'^{I}_{(x,y)}(\Theta_i))\}$ and σ represents the standard deviation estimator. The standard deviation estimator σ is computed as:

$$\sigma^2 = \frac{1}{\theta_a - 1}\sum_{i=1}^{\theta_a}(R'^{I}_{(x,y)}(\Theta_i))^2 - \frac{\theta_a}{\theta_a - 1}(\mu)^2 \tag{B.38}$$

   where μ and θ_a are the mean value and the 179 orientation values, respectively.

   In addition to the three texture features that are associated with the orientation of the auto-correlation function, we compute two other texture attributes which were first introduced

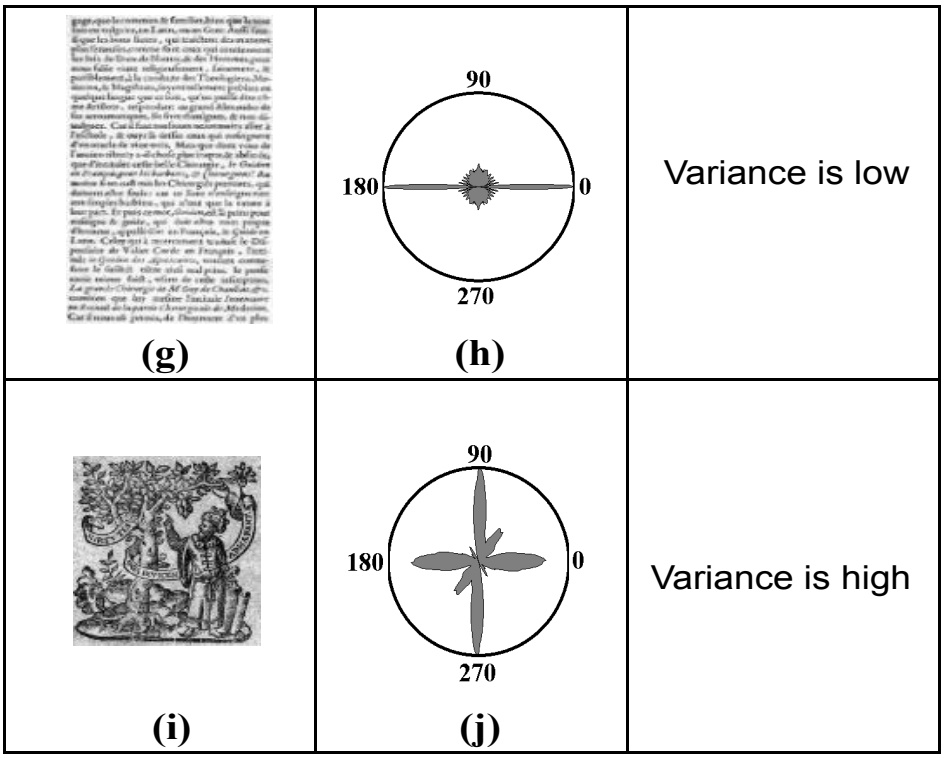


Figure B.15.: Examples of the variance of the intensities of the rose of directions. {(g) and (i)} are the original images and {(h) and (j)} are their rose of directions, respectively. The variance of intensities for the roses is high for graphic regions and low for text regions.

by Ouji *et al.* [272] and seem to be relevant for contemporary DIs and specifically with typographic characteristic characterization and chromatic/achromatic decomposition. The two texture descriptors are also related to the auto-correlation function through the mean stroke width and height of an image [272]. Ouji *et al.* computed these features in the horizontal and vertical directions [272]. In this work, we compute the mean stroke width and height along the axis of the main angle of the rose of directions to accurately estimate the main stroke thickness along specific directions.

4. ***Mean stroke width along specific directions***
The next texture index corresponds to the estimation of mean stroke width along specific directions $F^{(4)}_{(x,y)}$. It is deduced from a derivative of the auto-correlation function along the axis of the main angle of the rose of directions Θ (*cf.* equation B.35) if $\Theta \in [10, 80]$ (*cf.* equation B.39), otherwise the mean stroke width is estimated along the horizontal axis (*cf.* equation B.40). If the growth rate of the sequence S^{width} (*cf.* equations B.39 and B.40) is lower than 10%, we estimate the mean stroke width, otherwise we continue to compute the sequence S^{width} until we reach the horizontal borders of the sliding window. S^{width} is defined to be:

$$S^{width} = \sum_{\Theta \in [10,80]} |I(x,y) - T^{\Theta}_{(\alpha,0)}(I(\frac{y}{|\tan(\Theta)|}, y))| \tag{B.39}$$

$$S^{width} = \sum_{\Theta \in [0,9] \cup [81,180]} |I(x,y) - T^{\Theta}_{(\alpha,0)}(I(x,y))| \tag{B.40}$$

where $T^{\Theta}_{(\alpha,0)}(I(.,.))$ is the translation of the analysis window of an image I by α pixels along the axis of the main angle of the rose of directions $\Theta = F^{(1)}_{(x,y)}$.

5. ***Mean stroke height along specific directions***
The computation of the last texture attribute is similar to that of the fourth texture index $F^{(4)}_{(x,y)}$. $F^{(5)}_{(x,y)}$ is an estimation of the mean stroke height computed along the axis of the main angle of the rose of directions Θ (*cf.* equation B.35) if $\Theta \in [10, 80]$ (*cf.* equation B.41), otherwise the mean stroke height is estimated along the vertical axis (*cf.* equation B.42). If the growth rate of the sequence S^{height} defined in equations B.41 and B.42 is lower than 10%, the mean stroke height is estimated, otherwise we continue to compute the sequence S^{height} until we reach the vertical borders of the analyzed sliding window. S^{height} is defined to be:

$$S^{height} = \sum_{\Theta \in [10,80]} |I(x,y) - T^{\Theta}_{(0,\beta)}(I(x, x\ |\tan(\Theta)|))| \tag{B.41}$$

$$S^{height} = \sum_{\Theta \in [0,9] \cup [81,180]} |I(x,y) - T^{\Theta}_{(0,\beta)}(I(x,y))| \tag{B.42}$$

where $T^{\Theta}_{(0,\beta)}(I(.,.))$ is the translation of the analysis window of an image I by β pixels along the axis of the main angle of the rose of directions $\Theta = F^{(1)}_{(x,y)}$.

Figure B.16 illustrates the mean stroke width and height differences of two fonts (normal and bold text characters) along the axis of the main angle of the rose of directions. The estimation of mean stroke width (resp. height) along specific directions $F^{(4)}_{(x,y)}$ (resp. $F^{(5)}_{(x,y)}$) is defined according to the algorithm 8 (resp. 9). Some steps in the two algorithms are shown in red color (algorithms 8 and 9). This coloring is meant to highlight the main computation steps related to the particular angle ranges of the rose of directions used to estimate mean stroke width and height along specific directions.

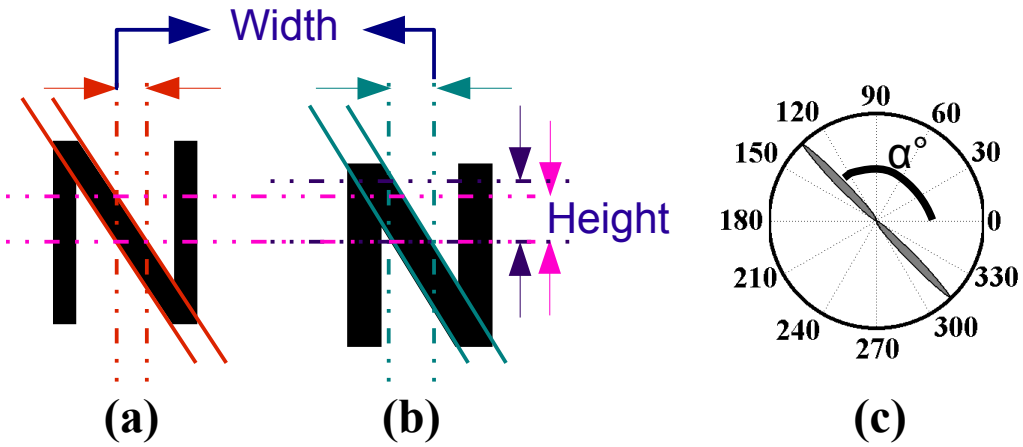


Figure B.16.: Estimation of the mean stroke width and height along specific directions. Figures {(a) and (b)} are the original images, and Figure (c) shows their rose of directions. Figure (a) depicts a normal text character, while Figure (b) illustrates a bold text character. As the main orientation of the rose of directions is oblique (*cf.* Figure (c)), the mean stroke width and height are estimated along the oblique axis.

Algorithm 8 Estimation of mean stroke width along specific directions

$pacc \leftarrow 0$
if $10 \leq \Theta \leq 80$ **then**
  $strokeWidth \leftarrow 1$
  while $strokeWidth < imageWidth$ **do**
    $acc \leftarrow 0$
    $y \leftarrow 0$
    while $y < imageHeight$ **do**
      $tacc \leftarrow 0$
      $x \leftarrow 0$
      $tx \leftarrow \left\lceil \frac{y}{|\tan(\Theta)|} \right\rceil - strokeWidth$
      while $x < imageWidth$ **do**
        $tacc \leftarrow tacc + |I(x, y) - I(tx, y)|$
        $x \leftarrow x + 1$
      $acc \leftarrow acc + tacc$
      $y \leftarrow y + 1$
    if $pacc \neq 0$ **then**
      $seqWidth \leftarrow \frac{acc - pacc}{pacc}$
      if $seqWidth \leq 0.1$ **then**
        return $strokeWidth$
    $pacc \leftarrow acc$
    $strokeWidth \leftarrow strokeWidth + 1$
  return $strokeWidth$
else
  $strokeWidth \leftarrow 1$
  while $strokeWidth < imageWidth$ **do**
    $acc \leftarrow 0$
    $y \leftarrow 0$
    while $y < imageHeight$ **do**
      $tacc \leftarrow 0$
      $x \leftarrow 0$
      while $x < imageWidth$ **do**
        $tx \leftarrow x - strokeWidth$
        $tacc \leftarrow tacc + |I(x, y) - I(tx, y)|$
        $x \leftarrow x + 1$
      $acc \leftarrow acc + tacc$
      $y \leftarrow y + 1$
    if $pacc \neq 0$ **then**
      $seqWidth \leftarrow \frac{acc - pacc}{pacc}$
      if $seqWidth \leq 0.1$ **then**
        return $strokeWidth$
    $pacc \leftarrow acc$
    $strokeWidth \leftarrow strokeWidth + 1$
  return $strokeWidth$

Algorithm 9 Estimation of mean stroke height along specific directions

```
1:  pacc ← 0
2:  if 10 ≤ Θ ≤ 80 then
3:      strokeHeight ← 1
4:      while strokeHeight < imageHeight do
5:          acc ← 0
6:          y ← 0
7:          while y < imageHeight do
8:              tacc ← 0
9:              x ← 0
10:             while x < imageWidth do
11:                 ty ← ⌈x|tan(Θ)|⌉ − strokeHeight
12:                 tacc ← tacc + |I(x, y) − I(x, ty)|
13:                 x ← x + 1
14:             acc ← acc + tacc
15:             y ← y + 1
16:         if pacc ≠ 0 then
17:             seqHeight ← (acc−pacc)/pacc
18:             if seqHeight ≤ 0.1 then
19:                 return strokeHeight
20:         pacc ← acc
21:         strokeHeight ← strokeHeight + 1
22:     return strokeHeight
23: else
24:     strokeHeight ← 1
25:     while strokeHeight < imageHeight do
26:         acc ← 0
27:         y ← 0
28:         while y < imageHeight do
29:             tacc ← 0
30:             x ← 0
31:             ty ← y − strokeHeight
32:             while x < imageWidth do
33:                 tacc ← tacc + |I(x, y) − I(x, ty)|
34:                 x ← x + 1
35:             acc ← acc + tacc
36:             y ← y + 1
37:         if pacc ≠ 0 then
38:             seqHeight ← (acc−pacc)/pacc
39:             if seqHeight ≤ 0.1 then
40:                 return strokeHeight
41:         pacc ← acc
42:         strokeHeight ← strokeHeight + 1
43:     return strokeHeight
```

B.1.5. GLCM features

The fifth set of texture features investigated in this work is the GLCM or co-occurrence attributes [180].

B.1.5.1. Generalities

The GLCM or co-occurrence matrix is a classic of statistical texture-based segmentation methods. The GLCM is an estimate of the second order probability density function of image pixels. This matrix determines the probability of occurrence of pixel pairs according to their gray-levels and distance by considering the spatial relationship of pixels in the image.

A GLCM element is the probability of the gray-level pairs defined in a specified direction θ_c and separated by a particular distance of d_c units (*cf.* Figure B.17). The co-occurrence descriptors are then statistics computed from the GLCM. They provide second order statistical information of neighboring pixels of an image. Multi-distance and multi-direction can be applied to extract a large number of GLCM descriptors.

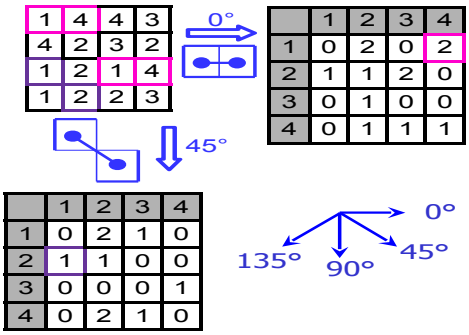


Figure B.17.: Illustration of the process of calculating the GLCM for the 0° and 45° directions.

B.1.5.2. State-of-the-art related to GLCM parametrization

Fourteen textural features extracted of the GLCM have initially been introduced by Haralick *et al.* [180] for texture discrimination of natural and satellite images which are widely known as the statistical Haralick features. The GLCM matrix was firstly used for the segmentation and representation of textures in images [572, 573]. Later, the use of the GLCM has been widespread in other fields of pattern recognition. For instance, in medicine the GLCM was computed from computed tomography images and several features were extracted for disease diagnostic [574]. Another example of using the GLCM was presented by Eleyan and Demirel [575]. They evaluated two methods of extracting feature vector from the GLCM for face recognition and classification. They stated that using the GLCM directly as the feature vector outperforms the feature vector containing the extracted Haralick features. A survey of DI segmentation methods using texture analysis presented different methods for segmenting DIs [173]. A novel texture analysis approach based on the assembly of n^{th} order co-occurrence information within a processing window was also proposed. This study stated that the GLCM approach is the best one in terms of processing time and complexity. For segmenting DI contents into text, graph, table and picture, Kim and Kim [175] analyzed six standard GLCM features (entropy, contrast, energy, uniformity, diagonal moment and homogeneity) in the entropy image.

A number of other works based on the GLCM feature extraction and analysis have also been proposed in order to segment and classify the content of DIs [274, 275]. More methods based on the GLCM feature analysis have been proposed in the literature for identifying script and language from DIs [276, 200]. For Arabic font recognition, the GLCM with $d_c = 4$ for 4 orientations $\theta_c = \{0°, 45°, 90°, 135°\}$ were used in [256]. Usually, the co-occurrence matrices are generated for a

small range of distance values $d_c = \{1, 2\}$ and typically for the directions $\theta_c = \{0°, 45°, 90°, 135°\}$ [200].

B.1.5.3. GLCM features

In this work, from the computed co-occurrence matrices, eight GLCM features are extracted for two distances $d_c = \{1, 2\}$ [274, 200]:

- Maximum entry in the GLCM or maximum probability (*cf.* equation B.43),
- Correlation metric (*cf.* equation B.44),
- Energy or angular second moment (*cf.* equation B.45),
- Entropy (*cf.* equation B.46),
- Inertia or contrast (*cf.* equation B.47),
- Local homogeneity (*cf.* equation B.48),
- Cluster shade (*cf.* equation B.49),
- Cluster prominence (*cf.* equation B.50).

In addition to the 16 co-occurrence features (eight for each distance), two other descriptors are computed (mean value (*cf.* equation B.51) and standard deviation (*cf.* equation B.52) of the energy) for the two combined distances [275]. The 18 extracted GLCM features have been shown to perform well for script identification in [200]. However, Haralick *et al.* [180] noted that it is hard to determine the textural characteristics and explain the accurate significance of each GLCM feature, the following extracted GLCM features in this work are meaningful:

1. ***Maximum probability***
   This metric ensures the record of the highest GLCM element. High values of GLCM element will occurred if one combination of pixels dominates pixel pairs. It is given by:

$$F_{d_c}^{(1)} = \max_{i,j} \left\{ p_{(d_c,\theta_c)}(i,j) \right\} \tag{B.43}$$

   where $p_{d_c,\theta_c}(i,j)$ is the probability of the gray-level pair i and j defined in a specified direction θ_c and separated by a particular distance of d_c units.

2. ***Correlation metric***
   This feature helps to measure the gray-level linear dependence between pixels at the specified positions relative to each other. It has a large value when the values are uniformly distributed in the GLCM and a low value otherwise. It is computed as:

$$F_{d_c}^{(2)} = \sum_{i=0}^{255} \sum_{j=0}^{255} \frac{(i - \mu_r)(j - \mu_c) p_{(d_c,\theta_c)}(i,j)}{\sigma_r \sigma_c} \tag{B.44}$$

   where

$$p_r(i) = \sum_{i=0}^{255} p_{d_c,\theta_c}(i,j) \qquad p_c(j) = \sum_{j=0}^{255} p_{d_c,\theta_c}(i,j)$$

$$\mu_r = \sum_{i=0}^{255} p_r(i) \qquad \mu_c = \sum_{j=0}^{255} p_c(j)$$

$$\sigma_r^2 = \sum_{i=0}^{255} i^2 p_r(i) - \mu_r^2 \qquad \sigma_c^2 = \sum_{j=0}^{255} j^2 p_c(j) - \mu_c^2$$

3. ***Energy***
   This measure which has also been called angular second moment, provides an insight of image homogeneity. It has low value when the probabilities of the gray-level pairs have very similar values and a high value otherwise. It is defined by:

$$F_{d_c}^{(3)} = \sum_{k=0}^{255} D(k) \tag{B.45}$$

   where $D(k) = \sum_{\substack{0 \le i \le 255 \\ |i-j|=k}} \sum_{0 \le j \le 255} p_{d_c,\theta_c}(i,j)$

4. ***Entropy***
   This metric characterizes the energy values for pixel combinations. It measures the disorder or randomness of the GLCM. Inhomogeneous texture have low first order entropy, while a homogeneous texture has a high entropy. It is defined by:

$$F_{d_c}^{(4)} = -\sum_{k=0}^{255} D(k) \log_2 D(k) \tag{B.46}$$

5. ***Contrast***
   This metric which has also been called inertia, corresponds to a measure of the contrast by computing a difference moment of the GLCM and it estimates the contrast or it quantifies local variation present in the analyzed image. It is defined by:

$$F_{d_c}^{(5)} = \sum_{k=0}^{255} k^2 D(k) \tag{B.47}$$

6. ***Local homogeneity***
   This measure has also been called inverse difference moment. It is higher when we find the same pair of pixels which is in the case that the gray-level is uniform or when there is a spatial periodicity. It is defined by:

$$F_{d_c}^{(6)} = \sum_{k=0}^{255} \frac{D(k)}{1+k^2} \tag{B.48}$$

7. ***Cluster shade***
   This metric corresponds to a measure of the gray-level distribution around the mean, with a high ability to discriminate the third order. It measures the skewness of the GLCM (*i.e.* lack of symmetry). When it is high, the analyzed image is not symmetric. It is defined by:

$$F_{d_c}^{(7)} = \sum_{i=0}^{255} \sum_{j=0}^{255} (i - M_r + j - M_c)^3 p_{(d_c,\theta_c)}(i,j) \tag{B.49}$$

8. ***Cluster prominence***
   This metric corresponds to a measure of the gray-level distribution around the mean, with a high ability to discriminate the fourth order. It also measures the skewness of the GLCM. It is defined by:

$$F_{d_c}^{(8)} = \sum_{i=0}^{255} \sum_{j=0}^{255} (i - M_r + j - M_c)^4 p_{(d_c,\theta_c)}(i,j) \tag{B.50}$$

9. ***Energy mean***
   This metric corresponds to the mean of the energy feature computed from the two distance values $d_c = 1, 2$. It is computed as:

$$F^{(17)}_{d_c=1,2} = \sum_{k=0}^{510} kD(k) \tag{B.51}$$

10. ***Energy standard deviation***
    This metric corresponds to the standard deviation of the energy feature computed from the two distance values $d_c = 1, 2$. It characterizes the uniformity of the texture when varying the specified distance. It is computed as:

$$F^{(18)}_{d_c=1,2} = \sqrt{\sum_{k=0}^{510} (k - F^{(13)}_{d_c=1,2})^2 D(k)} \tag{B.52}$$

B.1.6. Gabor features

The sixth set of texture features investigated in this work is the Gabor descriptors.

B.1.6.1. Generalities

The Gabor features are extracted using the multi-channel Gabor filtering technique. The original Gabor elementary functions have been firstly proposed by Gabor [277]. The multi-channel Gabor filtering is inspired by the multi-channel filtering theory which has been first investigated by Campbell and Robson [278] for the visual information processing of the human visual system. Daugman [279] modeled the visual information processing of the human visual system by the 2-D multi-channel Gabor functions which are local spatial band-pass filters. The main idea of the multi-channel filtering technique is to exploit the differences in dominant sizes and orientations of different textures by decomposing the original image into several filtered images with limited spectral information. The 2-D Gabor functions have the advantage to have the conjoint resolution information in both the 2-D spatial and Fourier domains. The filtered images are proceeded by tuning the analyzed image to combinations of frequency and orientation in a narrow range which are referred to channels and interpreted as band-pass filters. By applying a bank of GFs, the specified channels cover the spatial-frequency domain. Ursani *et al.* [576] presented an empirical comparison between texture features based on the discrete Fourier transform and GFs for texture recognition and retrieval. They proved that analyzing the Gabor features in image datasets containing noisy and rotated variants of texture performs better than analyzing the Fourier descriptors for texture recognition and retrieval. Hence, GFs have been shown to have good performance, due to its optimal localization properties to capture information in both the spatial and frequency domains from the analyzed images, as opposed to the Fourier transform.

A 2-D GF is a linear selective band-pass filter, dependent on two parameters (spatial frequency f_g and orientation θ_g) which characterize the specified channel. It consists of a Gaussian kernel function modulated by a sinusoidal plane wave. The spatial frequency f determines the distance from the Gaussian centers to the origin while the orientation θ_g specifies the angle from the horizontal axis (*i.e.* α-axis to the Gaussian centers). The multi-channel Gabor filtering approach is inherently multi-resolutional which is a close relative of the wavelet transform [218].

The Gabor transform of an image $I(x, y)$ is:

$$I_{G_{(f_g,\theta_g)}}(x, y) = \sum_{\alpha\in\Omega} \sum_{\beta\in\Omega} I(x + \alpha, y + \beta)\ G_{(f_g,\theta_g)}(\alpha, \beta) \tag{B.53}$$

where f_g and θ_g are the spatial frequency and orientation of the Gabor filter envelope.

$$G_{(f_g,\theta_g)}(\alpha,\beta) = \sqrt{[G_{e(f_g,\theta_g)}(\alpha,\beta)]^2 + [G_{o(f_g,\theta_g)}(\alpha,\beta)]^2}$$

$$G_{e(f_g,\theta_g)} = \frac{H_{1(f_g,\theta_g)}(\alpha,\beta) + H_{2(f_g,\theta_g)}(\alpha,\beta)}{2}$$

$$G_{o(f_g,\theta_g)} = \frac{H_{1(f_g,\theta_g)}(\alpha,\beta) + H_{2(f_g,\theta_g)}(\alpha,\beta)}{2j}$$

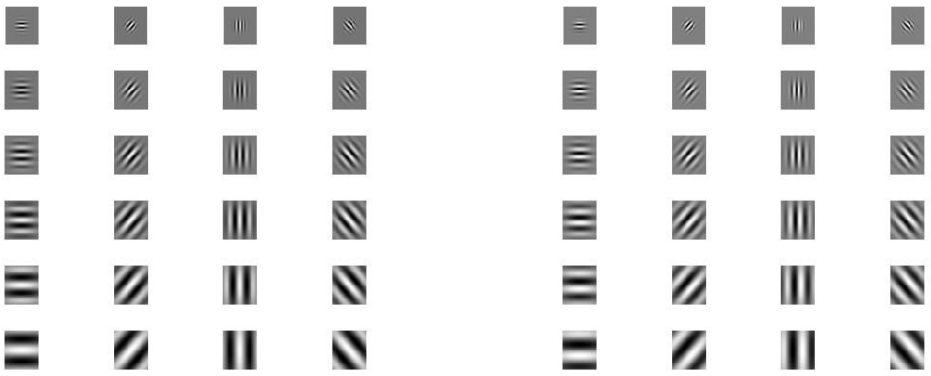
$$H_{1(f_g,\theta_g)}(\alpha,\beta) = \exp\{-2\pi\sigma_g^2[(\alpha - f_g\cos\theta_g)^2 + (\beta - f_g\sin\theta_g)^2]\}$$

$$H_{2(f_g,\theta_g)}(\alpha,\beta) = \exp\{-2\pi\sigma_g^2[(\alpha + f_g\cos\theta_g)^2 + (\beta - f_g\sin\theta_g)^2]\}$$

$$j^2 = -1$$

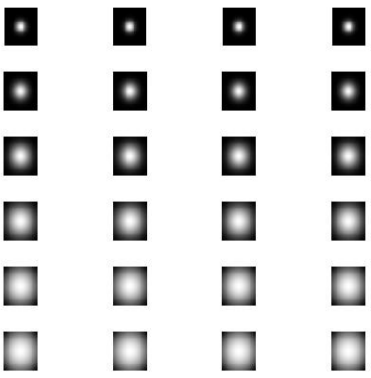
where $G_{e(f_g,\theta_g)}$ and $G_{o(f_g,\theta_g)}$ denote the spatial frequency responses of the even- and odd- symmetric GF. σ_g denotes the space constant of the Gabor filter envelope.

An illustrative example of the real parts, imaginary parts and magnitudes of 24 GFs (6 different spatial frequencies f_g=$\{2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}$ and $64\sqrt{2}\}$ and 4 different orientations θ_g=$\{0, \pi/4, \pi/2$ and $3\pi/4\}$) is presented in Figure B.18.



(a) Real parts of GFs

(b) Imaginary parts of GFs



(c) Magnitudes of GFs

Figure B.18.: Illustration of the real parts, imaginary parts and magnitudes of GFs (6 different spatial frequencies f_g=$\{2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}$ and $64\sqrt{2}\}$ and 4 different orientations θ_g=$\{0, \pi/4, \pi/2$ and $3\pi/4\}$).

B.1.6.2. State-of-the-art related to Gabor parametrization

Texture features generated by GFs have been increasingly considered and applied to DIA. During the last two decades, Gabor-based analysis approaches have been proposed for biometric identification based on handwriting [280, 156, 281], writer identification [282], handwritten word recognition [283], character recognition [284], font recognition [285], script identification [286, 287], signature recognition [288], palm print recognition [289], degraded DI binarization [290], *etc.* Zhu *et al.* [285] proposed a texture-analysis-based algorithm for automatic font recognition by extracting the Gabor features. They noted a $99,1\%$ of mean recognition rate. Ma and Doermann [257] proposed a GF-based multi-class classifier in order to identify scripts, and font faces and styles. A binarization method based on Gabor filter bank for ancient degraded DIs was proposed in [290]. A GF bank with four orientations (0, $\pi/4$, $\pi/2$ and $3\pi/4$) weighted by the dominant foreground script slant angle of the DI and one selected frequency was used to determine more efficiently the foreground information.

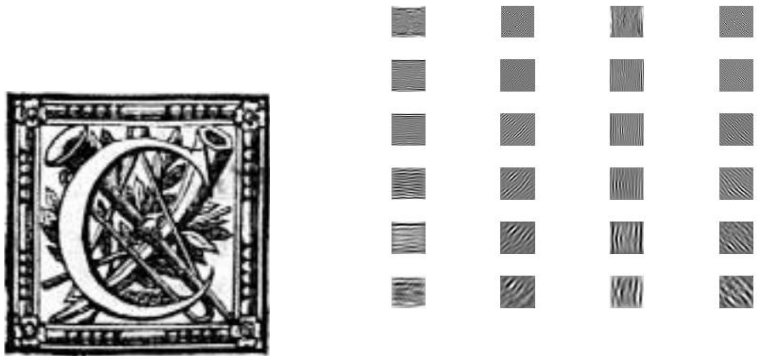
Nevertheless, numerous approaches have been sought for text segmentation and extraction from digital DIs using the Gabor descriptors [189, 291, 292, 191]. Several studies have been conducted in the literature for page layout analysis using the multi-channel GFs [293, 257, 294], while few ones have explored GFs for HDI segmentation. For instance, Ribeiro *et al.* [237] proposed an optical character recognition (OCR) system for HDI analysis and recognition by applying fuzzy methods on aligned oriented features extracted using GFs in the training step. Vieux and Domenger [216] proposed a pixel-based classification approach to separate text from other classes (e.g. illustrations and background) by using a bank of GFs at five scales (1, $\sqrt{2}$, 2, $2\sqrt{2}$ and 4) and six orientations ($k\frac{\pi}{6}, k \in \{0,\ldots,5\}$). Their approach was evaluated on a public dataset containing magazines and technical journals. They found 86%, 82.7% and 53.7% of F-measure for segmenting background, text and graphic pixels, respectively. Jain *et al.* [248] showed the effectiveness of applying a multi-channel Gabor filtering-based texture segmentation approach for segmentation and classification of DIs. They chose the five following spatial frequencies: $4\sqrt{2}$, $8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$ and $64\sqrt{2}$. Charrada and Ben Amara [238] extracted nets from ancient Arab periodicals by exploring GFs. Zhong and Cheriet [239] used the dimensionally reduced multi-channel GFs for text block identification on image patches from HDIs. They extracted 28 GFs from image patches in their experiments, where 7 spatial frequencies ($\sqrt{2}$, $2\sqrt{2}$, $4\sqrt{2}$, $8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$ and $64\sqrt{2}$) and 4 orientation angles (0, $\pi/4$, $\pi/2$ and $3\pi/4$) were pre-defined. Cruz-Fernández and Ramos-Terrades [64] computed a 36-D Gabor feature vector for each analyzed pixel using 9 orientations (0, $2\pi/9$, $4\pi/9$, $6\pi/9$, $8\pi/9$, $10\pi/9$, $12\pi/9$, $14\pi/9$ and $16\pi/9$) and 4 spatial frequencies (an overlapping degree of 0.5 in the frequency domain with the highest frequency is equal to 0.35) for structured HDI segmentation. For Arabic font recognition, 16 Gabor channels were computed with 4 frequencies $f_g = \{8, 16, 32, 64\}$ and 4 orientations $\theta_g = \{0°, 45°, 90°, 135°\}$ in [256]. A learning-free approach to detect the main text area from side-notes in ancient manuscripts based on coarse-to-fine scheme [240]. A coarse segmentation of the main text area was processed by using GFs. The proposed approach achieved promising results in terms of segmentation quality (*i.e.* 98.84% of mean F-measure was noted on 38 HDIs) and time performance (*i.e.* $01'\ 13''$ per page on average). The four directions (0, $\pi/4$, $\pi/2$ and $3\pi/4$) are widely used in the literature [189, 248, 285, 257].

Designing the proper channels in order to generate filters tuned to several different frequencies and orientations has been illustrated as the crucial issue in using GFs for texture characterization. Dunn *et al.* [577, 578] suggested an automatic approach for finding the optimal channels for discriminating different textures, but the computational complexity is very high. Bianconi and Fernández [579] evaluated the impact of the GF parameters on texture classification. They reported that an increase of the number of frequencies and orientations has an insignificant influence on texture classification performance. But, they confirmed that the best performance of texture classification is conditioned by the design of the convenient Gabor channels. Clausi and Jernigan [580] presented a comparative study of different techniques used to extract the Gabor descriptors for texture discrimination. They showed that the magnitude response outperforms the other different evaluated methods, such as

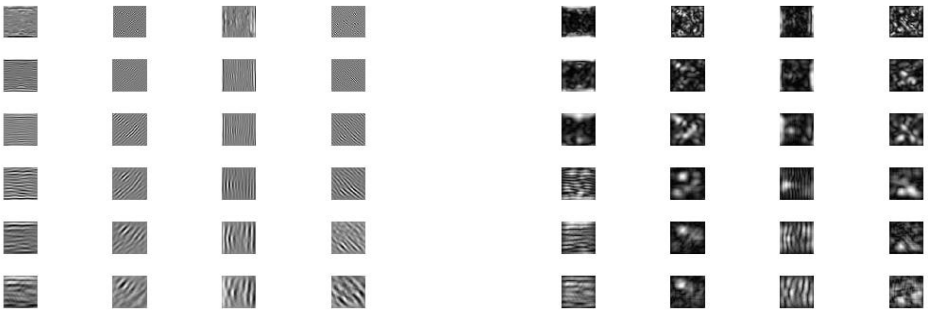
using only the real component, *etc.* Arivazhagan *et al.* [581] introduced the rotation invariant features by computing the mean and variance of the Gabor filtered image for texture classification.

B.1.6.3. Gabor features

In this work, the magnitude response of the output of Gabor functions is investigated. The magnitude of the output is important if the specified GF matched the particular texture, otherwise low response to the specified GF corresponds to poor match of the dominant texture properties of the analyzed image to the set of the spatial-frequency components of the fixed GF [295]. 24 GFs are applied (6 different spatial frequencies f_g=$\{2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}$ and $64\sqrt{2}\}$ and 4 different orientations θ_g=$\{0, \pi/4, \pi/2$ and $3\pi/4\}$) (*cf.* Figure B.18). The space of GF is set constant $\sigma_g = \sigma_x = \sigma_y = 1$. When convolving an image with 24 Gabor channels (obtained by using 6 different spatial frequencies and 4 different orientations), 24 Gabor filtered images are produced (*cf.* Figure B.19). In this work, 24 responses of filtered images or Gabor responses are generated (*cf.* Figure B.19(d)).



(a) Input image (b) Real parts of the 24 Gabor filtered images



(c) Imaginary parts of the 24 Gabor filtered images (d) Magnitudes of the 24 Gabor filtered images

Figure B.19.: Illustration of the real parts, imaginary parts and magnitudes of 24 Gabor filtered images obtained after applying 24 GFs (6 different spatial frequencies f_g=$\{2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2}$ and $64\sqrt{2}\}$ and 4 different orientations θ_g=$\{0, \pi/4, \pi/2$ and $3\pi/4\}$) on a drop cap image.

Finally, by convoluting the analyzed whole DI at each specified channel defined by a pair of orientation and frequency, the Gabor features are extracted from the magnitudes of the Gabor filtered images (*cf.* Figure B.19(d)). The extracted Gabor features represent the statistical distribution of the Gabor magnitude response. They consist of two simple statistics: the mean value (*cf.* equation B.54) and standard deviation (*cf.* equation B.55) of the Gabor filtered magnitude response corresponding to all pixels defined in the analyzed sliding window of the filtered image.

1. ***Mean of the Gabor filtered magnitude response***
   This feature characterizes the average of the Gabor filtered magnitude response corresponding to all pixels defined in the analyzed sliding window of the filtered image. This descriptor quantifies how the dominant texture properties of the analyzed image match to the set of spatial-frequency components of the fixed GF. It is given by:

$$F^{(1)}_{(f_g,\theta_g)} = \frac{\sum_{x=1}^{M_g}\sum_{y=1}^{N_g} I_{G_{(f_g,\theta_g)}}(x,y)}{M_g\ N_g} \tag{B.54}$$

   where M_g and N_g denote the width and height of the Gabor filtered magnitude response, respectively.

2. ***Standard deviation of the Gabor filtered magnitude response***
   This descriptor determines how much the dispersion from the computed mean of the Gabor filtered magnitude response exists. It is given by:

$$F^{(2)}_{(f_g,\theta_g)} = \frac{\sum_{x=1}^{M_g}\sum_{y=1}^{N_g} [I_{G_{(f_g,\theta_g}}(x,y) - F^{(1)}_{(f_g,\theta_g)}]^2}{M_g\ N_g} \tag{B.55}$$

B.1.7. Wavelet features

The last set of textural features examined in this work is the wavelet descriptors.

B.1.7.1. Generalities

Mallat [154] investigated the application of the wavelets as multi-resolution representations to data compression in image coding, texture discrimination and fractal analysis. The wavelet features which are extracted from the wavelet transform provide interesting insight on the statistical characteristics of the analyzed image. The wavelet features represent consistent properties in the localization of the frequency space and multi-resolution.

A 2-D wavelet transform ensures the localization in both the scale (frequency) domain via dilations and in the time domain via translations of the mother wavelet. A 2-D wavelet transform represents an image with both the spatial and frequency characteristics. The 2-D wavelet decomposition is processed by using a high-pass filter g^f, a low-pass filter h^f and a 2-D scaling function ϕ and by assuming the three following wavelet functions:

$$\begin{cases} \psi^{(I)}(x,y) = \phi(x)\psi(y) = 2\sum_{k,l} g^{(I)}(k,l)\phi(2x-k,2y-l) \\ \psi^{(II)}(x,y) = \psi(x)\phi(y) = 2\sum_{k,l} g^{(II)}(k,l)\phi(2x-k,2y-l) \\ \psi^{(III)}(x,y) = \psi(x)\psi(y) = 2\sum_{k,l} g^{(III)}(k,l)\phi(2x-k,2y-l) \end{cases} \tag{B.56}$$

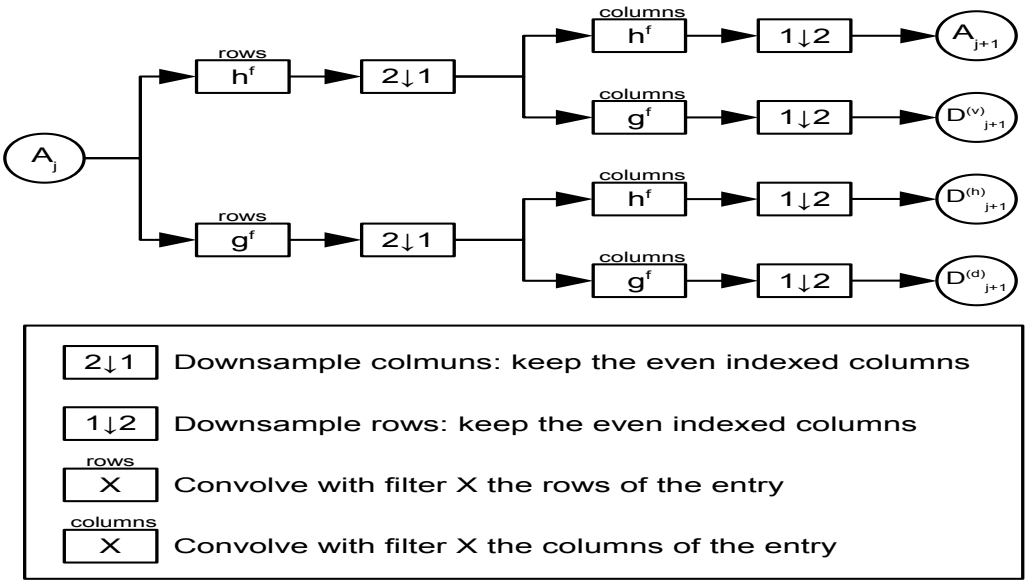
where ψ denote a wavelet function.

$$\begin{cases} g^{(I)}(k,l) = & h^f(k)g^f(l) \\ g^{(II)}(k,l) = & g^f(k)h^f(l) \\ g^{(III)}(k,l) = & g^f(k)g^f(l) \end{cases} \tag{B.57}$$

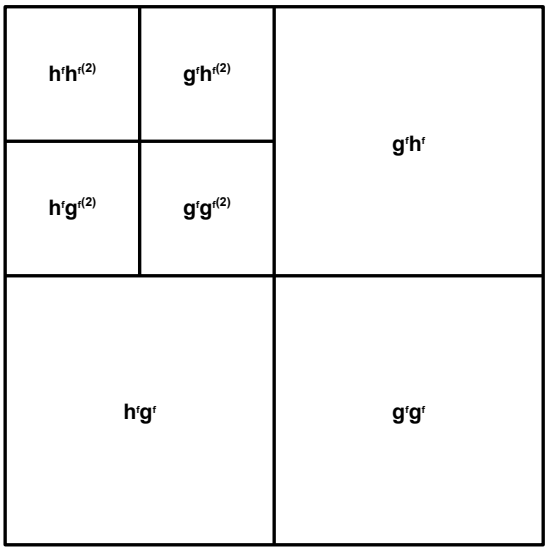
The objective of a 2-D wavelet transform is to decompose an image into low and high frequency sub-band images (*i.e.* to filter out several frequencies range). The 2-D J-level wavelet transform decomposes a discrete input image $I(x, y)$ into 4 sub-bands and it produces $3J + 1$ sub-images:

$$A_{2^{-J}}, \{D^{(v)}_{2^{-j}}, D^{(h)}_{2^{-j}}, D^{(d)}_{2^{-j}}\}_{j=1,2,\ldots,J}$$

where J represents the scale of the discrete wavelet transform. j denotes the decomposition level of the discrete wavelet transform such as $j = 1, 2, \ldots, J$. $A_{2^{-J}}$ is the approximation of the input image $I(x, y)$ at 2^{-J} resolution. $D^{(v)}_{2^{-j}}$, $D^{(h)}_{2^{-j}}$ and $D^{(d)}_{2^{-j}}$ are 3 detail components of the input image $I(x, y)$ at 2^{-j} resolution. The wavelet coefficients in $D^{(v)}_{2^{-j}}$, $D^{(h)}_{2^{-j}}$ and $D^{(d)}_{2^{-j}}$ illustrate the vertical, horizontal and diagonal high frequencies, respectively (*cf.* Figure B.20).



(a) Filter bank structure of the 2-D wavelet decomposition



(b) 2-D 2-level wavelet transform

Figure B.20.: Illustration of the 2-D wavelet decomposition.

The approximation (*cf.* equation B.58) and detail (*cf.* equation B.59) coefficients are computed according to the following equations:

$$C_{k,l}^{Aj} = \int_{-\infty}^{+\infty} 2^j \phi(2^j x - k, 2^j y - l) f_s(x, y) dx dy \tag{B.58}$$

$$C_{k,l}^{D(s)j} = \int_{-\infty}^{+\infty} 2^j \psi^{(s)}(2^j x - k, 2^j y - l) f_s(x, y) dx dy \tag{B.59}$$

where $(s)j$ denotes the vertical, horizontal or diagonal detail components of the input image $I(x, y)$ at 2^{-j} resolution. $f_s(x, y)$ represents the pixel gray-level of a sub-band or sub-image from the 2-D wavelet decomposition.

B.1.7.2. State-of-the-art related to wavelet parametrization

First, the wavelet transform has been developing as one of the most powerful processors in various applications of signal and image processing (e.g. compression, enhancement, analysis, classification, detection and recognition) [582, 583]. For instance, Lee [584] proposed a family of 2-D Gabor wavelets based on the Daubechies wavelet transform for a complete image representation. There has been a large number of wavelets for both continuous and discrete analysis. Coiflet, Morlet, complex Morlet, Mexican Hat, Symlet, B-spline bi-orthogonal, derivative of Gaussian, complex Gaussian, discrete approximation of Meyer, Shannon, frequency B-spline, Paul, Haar, Daubechies and Cohen-Daubechies-Feauveau are examples of wavelet families. The choice of the family wavelet is dictated by the nature of the application and image characteristics. The different wavelet families vary according to several properties. To name a few, the wavelet support in time and frequency, wavelet symmetry or anti-symmetry, regularity wavelet, existence or not of a scaling function, *etc.* The number of taps denotes the number of coefficients in the wavelet filter. For instance, Haar, Db3 and Db4 wavelets have 2, 6 and 8 taps, respectively.

Even if the wavelet transform is computationally expensive (*i.e.* it is carried out by a large combination of filter parameters), it has been proved to be a promising alternative of many texture approaches such as GFs for many fields of computer vision and pattern recognition [585]. For texture-based image retrieval, the wavelet-based approaches have been proposed [143, 585, 315]. Ben Abdeljelil *et al.* [586] designed a compactly supported orthonormal wavelet for image denoising and compression. The Haar wavelets have been applied for biomedical image segmentation [587]. Traina *et al.* [588] proposed an application allowing to index and retrieve medical images based on the wavelet features. Boukhris *et al.* [583] extracted textural features from the Daubechies wavelet transform for the artificial human face recognition. Myint *et al.* [589] evaluated four different wavelet decomposition procedures which were performed up to 3 levels to digitally classify urban land use and land cover categories using high resolution images. Svensson *et al.* [590] evaluated several 2-D wavelet filters, such as Daubechies wavelets, for the estimation of differences in textures of pharmaceutical tablets. Moesa *et al.* [370] used the discrete wavelet transform to smooth the noise in the gene expression dataset.

Since the wavelet transform has been an attractive tool and has provided interesting results for image characterization, several wavelet derivatives have been designed to improve the performance of texture segmentation and classification. For instance, Van de Wouwer *et al.* [591] introduced two feature sets: the wavelet histogram signatures and wavelet co-occurrence signatures generated from the discrete wavelet transform for a statistical characterization of textures. Laine and Fan [592] introduced a generalization of orthonormal and compactly supported wavelets, namely the wavelet packets, for a texture characterization at multiple scales. Unser [593] proposed a fast approach based on discrete wavelet frames for texture classification and segmentation. Etemad and Chellappa [594] proposed a class separability measure as a wavelet feature based on the wavelet packet trees for texture classification. Chang and Kuo [406] applied a tree-structured wavelet transform for texture

classification and segmentation. Another application of the wavelet transform for texture analysis consisted of extracting other texture descriptors, such as the LBP features from the sub-bands resulting from the wavelet transform [595].

In a growing number of areas, the wavelet-based methods have been investigated more and more. Recently, a lot of studies of applying the wavelet transform have been reported for many fields of DIA. The wavelet transform has been very effective for DI pre-processing [296], watermarking [208], handwriting-based writer identification [201], script identification [200, 255], text localization [217, 297], page segmentation [212], printer discrimination [207], *etc.* Maatouk *et al.* [208] showed that the 3-level decomposition with the Db2 and Db3 family provided the best performance for the watermarking of HDIs. Kricha *et al.* [296] proposed a denoising step by applying a thresholding technique in the coefficients of wavelet sub-bands to reduce the noise in the background of HDIs. Furukawa [207] used the bi-orthogonal spline 2 wavelet transform for discriminating printers based on contours qualities of printed characters. For script recognition, Busch *et al.* [200] evaluated a number of wavelet features based on energy, logarithmic mean deviation, logarithmic co-occurrence and scale co-occurrence. Baâti *et al.* [255] used the energy of 12-level bi-orthogonal wavelet coefficients for script identification. Hiremath and Shivashankar [298] also extracted features from the co-occurrence histograms of wavelet decomposed images for script identification. They concluded that the Haar wavelet yields the best classification results. Manthalkar *et al.* [299] also computed the rotation and scale invariant texture features using the discrete wavelet packet transform for script identification. They evaluated two wavelet families (bi-orthogonal and Daubechies) and they concluded that the bi-orthogonal wavelet outperforms the Daubechies ones (*i.e.* 83.07% and 80.89% of overall correct classification for the bi-orthogonal and Daubechies wavelets, respectively). Pardeshi *et al.* [222] extracted the directional multi-resolution information based on the Daubechies9 wavelet transform to automatically identify automatic handwritten Indian scripts. For the handwriting-based writer identification, He *et al.* [201] used the 3-level wavelet transform using a 4-tap Daubechies filter. Many studies applied the 3-level wavelet transform by using a 3-tap Daubechies filter to identify Arabic font [300, 301, 302, 303]. Gazzah and Ben Amara [304] explored the 2-D discrete wavelet transform based on a lifting scheme for writer identification (off-line Arabic handwriting). They compared 9 wavelet families, including the three following Daubechies wavelets (Daubechies2, Daubechies3 and Daubechies5), 4 Cohen-Daubechies-Feauveau wavelets, lazy wavelet transform and Symlet wavelets. They reported that the different evaluated wavelets give similar results (equal to 95%). He *et al.* [305] compared GFs with a novel wavelet approach based on the generalized Gaussian density for the off-line handwriting-based writer identification. They showed that the proposed approach based on the wavelet transform performs better than the traditional 2-D GFs and it is better in terms of the processing time. Ding *et al.* [306] used the 3-level spline2 wavelet transform on the normalized image of a single Chinese character for the character independent font recognition. Zhang *et al.* [307] performed a statistical analysis on the stroke patterns obtained from the wavelet decomposed sub-images using a 2-tap Symlet filter for the italic font recognition. For Arabic font recognition, the wavelet energy (*i.e.* sum of square of the detailed wavelet transform coefficients) was extracted from the Daubechies2 wavelet transform in [256]. Angadi and Kodabagi [308] extracted texture features (the zone wise wavelet energy features, vertical run statistical features of the wavelet coefficients and wavelet logarithmic mean deviation) from the wavelet transform for the word level script identification of text in the low resolution display board images. For multi-font Arabic character analysis and the extraction and classification of the handwritten shapes from ancient manuscripts, derivative forms of the wavelet transforms (e.g. ridgelet, curvelet and contourlet transforms) have been used [309, 241]. These specific wavelets offer the best trade-off between local and global features for handwritten recognition.

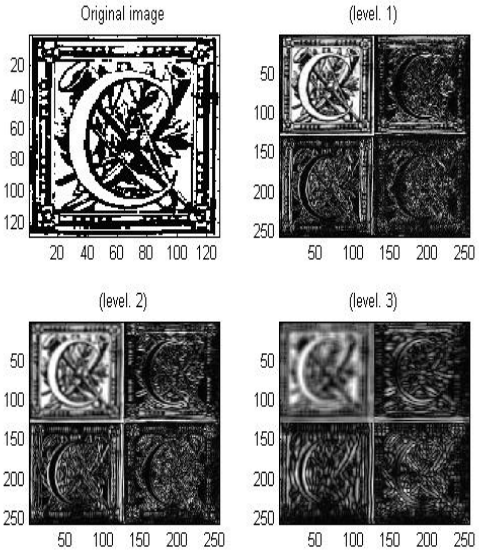
For page segmentation, Gupta *et al.* [212] studied the energy distribution over different scales of the orthonormal wavelet decomposition. Li and Gray [219] investigated the distribution characteristics of the wavelet coefficients of the 1-level Haar transform for DI segmentation. They noted that the results produced by the two longer wavelet filters (4-tap Daubechies and 8-tap Daubechies) are

similar while the Haar transform has the best localization property since its filter is the shortest and it has the least processing time. They extracted two novel features related to the pattern distribution of the wavelet coefficients using the Haar wavelet transform instead of computing moments of the wavelet coefficients as features. The first descriptor defines the rate of fit goodness of the distribution of the wavelet coefficients in high frequency bands to the Laplacien distribution. Then, the second feature determines the concentration rate of the wavelet coefficients in high frequency bands at few discrete values. They noted a 4.1% of average classification error rate. Kumar *et al.* [310, 217] compared the Haar discrete wavelet transform and matched wavelet for text extraction and DI segmentation. Liang and Chen [297] suggested to use the Haar discrete wavelet transform for the text region extraction from the static images or video sequences. They showed an average error rate close to 1.42%. Acharyya and Kundu [311] presented a multi-scale analysis method based on the wavelet scale-space features using a 8-tap filter for the text segmentation in DIs. Nourbakhsh *et al.* [2] used the log-polar wavelet energy signatures for the text localization and extraction from the complex gray-scale DIs. Jin and Tang [312] proposed a novel approach to determine the positions of the text areas in the complex-background images using the wavelet decomposition. Etemad *et al.* [249] presented an algorithm based on the pyramidal wavelet transform and wavelet packet tree using the Daubechies filters for the segmentation of unstructured DIs. A wavelet-based technique has been proposed for the reference line extraction from gray-level background DIs in [313]. For the text/non-text segmentation in DIs, Deivalakshmi *et al.* [314] extracted the wavelet-based GLCM features. The evaluated wavelet transforms are: Haar, Db4, Db25, Symlet8, Coilflet3 and Coilflet5. The Coilflet5 wavelet transform used in their algorithm outperforms the five other investigated wavelets. An average classification rate equal to 92.97% has been obtained with using the Coilflet5 filter. Kricha and Ben Amara [242] explored the correlation between the different sub-bands of the same decomposition level and the auto-correlation of each sub-band in the wavelet transform for the text/graphic separation in HDIs and the discrimination of the different alphabet kinds (Arabic, Latin and Hebrew). They computed the 1-order and 2-order statistics performed from the correlation function of each analysis window. Subsequently, they took into consideration only the mean and standard deviation of the auto-correlation of the approximation sub-band obtained from the 3-level decomposition of the wavelet transform and performed at four different sizes of analysis windows in order to adopt a multi-scale approach.

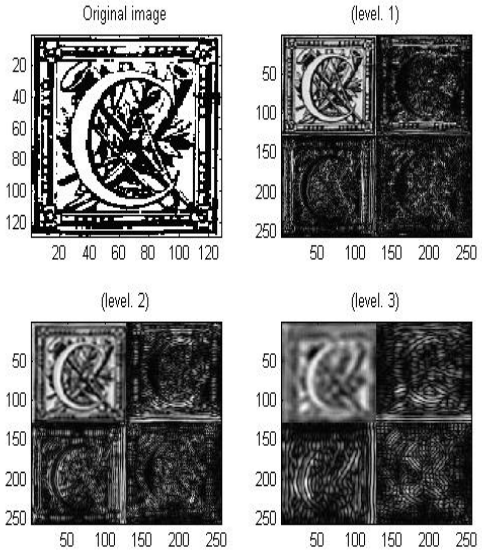
The Haar and Daubechies wavelets are the most used ones since they have been proved to work effectively in many applications. The Haar wavelet transform is the fastest among all wavelets since its coefficients are either 1 or -1. Thus, they are the less complex, simplest and most widely used wavelets, while the Daubechies ones are characterized by the fractal structures [297, 315]. Albuz *et al.* [596] computed the sum of squares of the wavelet coefficients of each sub-band for their image retrieval system. Myint *et al.* [589] computed four feature measures (log energy, Shannon's index, entropy and angular second moment) for the texture characterization of the urban land use and land cover classes. Nevertheless, the 1-order and 2-order statistics of the sub-band coefficients, such as the mean, energy and standard deviation of the wavelet coefficients, are the most commonly used features for the texture classification and segmentation issues [591]. Sheikholeslami *et al.* [597] extracted the mean and variance of the wavelet coefficients to characterize the image contrast. Laine and Fan [592] extracted the energy and entropy metrics for each wavelet packet for the characterization of textures in images. Busch *et al.* [598] proposed a logarithmic quantization of the wavelet coefficients to improve the texture classification performance. Myint *et al.* [253] demonstrated that the classification accuracy decreased when the wavelet decomposition level is high for the urban spatial feature discrimination. Angadi and Kodabagi [308] stated that the wavelet coefficients are the most suitable for the representation of textures in images. Kautsky *et al.* [599] reported that the wavelet transforms with an important number of taps are more suitable for the images without neither sharp edges nor many details. They concluded that the shorter wavelets (*i.e.* with a limited number of taps) perform better on the images which are characterized by a dominance of high-frequency information.

B.1.7.3. Wavelet features

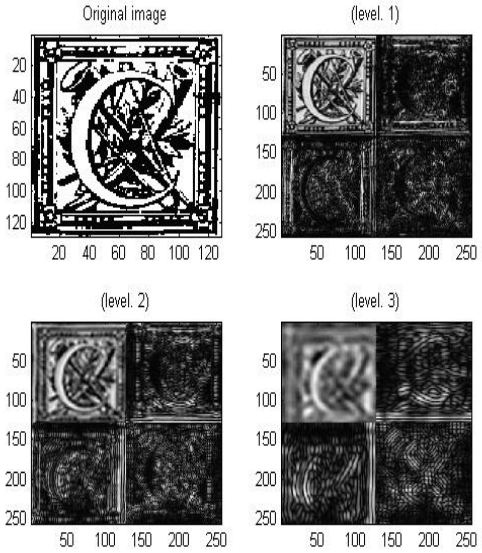
In this work, the wavelet features are extracted from the 2-D 3-level discrete stationary wavelet transform with a limited number of taps (3-level wavelet transform using Haar filter (Haar), 3-level wavelet transform using 3-tap Daubechies filter (Db3) and 3-level wavelet transform using 4-tap Daubechies filter (Db4)) (*cf.* Figure B.21). Therefore, 10 sub-bands ($A_{2^{-3}}$, $D^{(v)}_{2^{-1}}$, $D^{(h)}_{2^{-1}}$, $D^{(d)}_{2^{-1}}$, $D^{(v)}_{2^{-2}}$, $D^{(h)}_{2^{-2}}$, $D^{(d)}_{2^{-2}}$, $D^{(v)}_{2^{-3}}$, $D^{(h)}_{2^{-3}}$ and $D^{(d)}_{2^{-3}}$) are generated (*cf.* Figure B.22).



(a) Haar



(b) Db3



(c) Db4

Figure B.21.: Illustration of the application of 2-D 3-level discrete stationary wavelet transforms (Haar, Db3 and Db4) on a drop cap image.

In our experiments, in order to reduce the number of the wavelet coefficients, two simple statistics deduced from the wavelet transform coefficients for each sub-band are extracted to form feature vector of 20 terms (10 sub-bands). They represent the statistical distribution of the wavelet co-

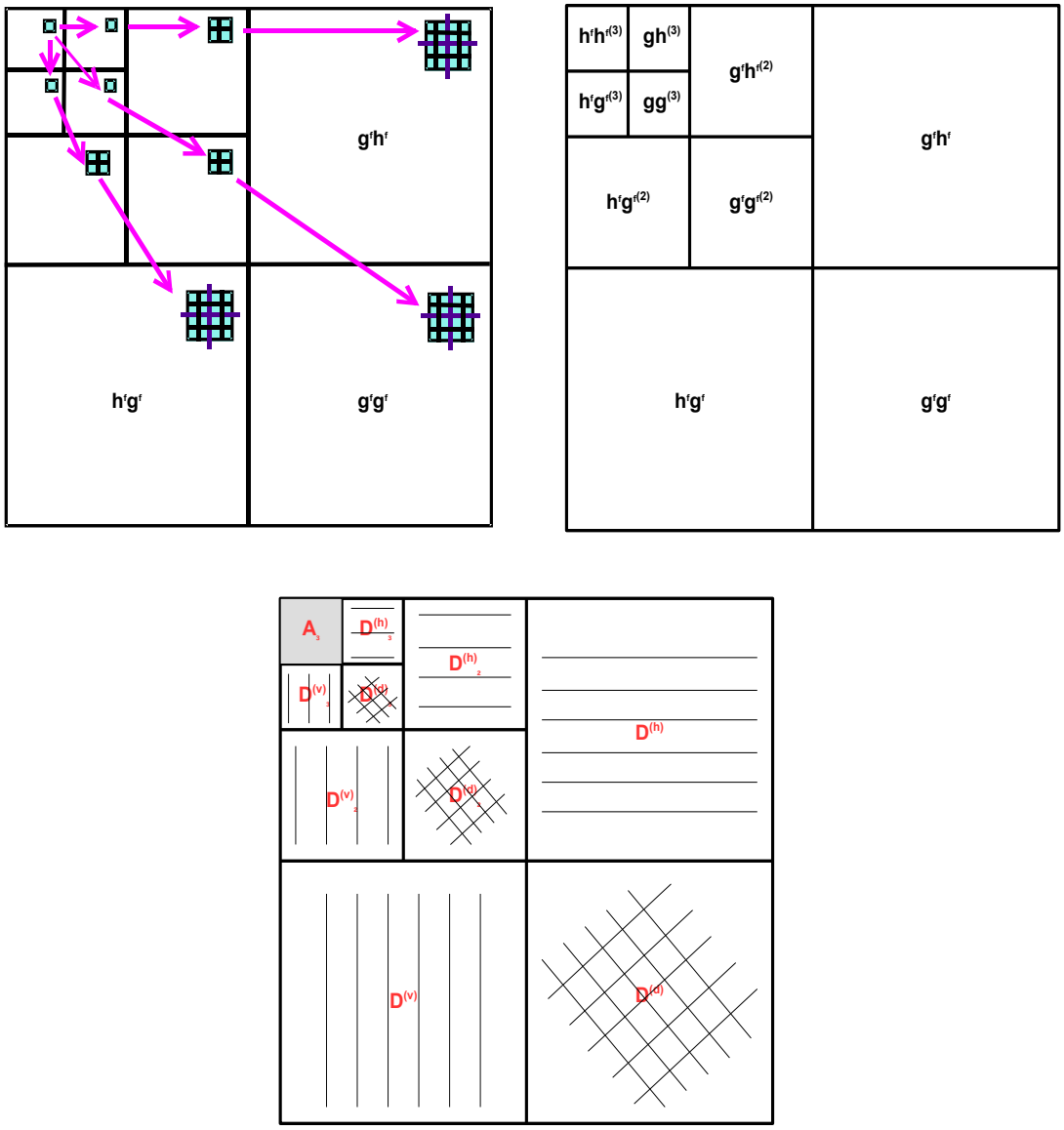


Figure B.22.: Illustration of the 2-D 3-level wavelet transform.

efficients. The two simple statistics: the mean value (*cf.* equation B.60) and standard deviation (*cf.* equation B.61) of the wavelet transform coefficients for each sub-band defined in the analyzed sliding window of the image, are extracted.

1. ***Mean of the wavelet transform coefficients***
   This feature characterizes the average of the wavelet transform coefficients for each sub-band defined in the analyzed sliding window of the image. This descriptor represents the average of 2-D signal in various frequency bands. It is computed as:

$$F^{(1)} = \frac{\sum_{i=0}^{S_w} \sum_{j=1}^{S_h} C(i,j)}{S_w \; S_h} \tag{B.60}$$

   where $C(i,j)$ is the transform wavelet coefficient. S_w and S_h are the width and height of a sub-band in the wavelet domain, respectively.

2. ***Standard deviation of the wavelet transform coefficients***
   This descriptor determines how much the dispersion from the computed mean of wavelet transform coefficients exists. It is computed as:

$$F^{(2)} = \frac{\sum_{i=0}^{S_w} \sum_{j=1}^{S_h} [C(i,j) - F^{(1)}]^2}{S_w \ S_h} \tag{B.61}$$

Three kinds of wavelet transform are assessed in this work, 3-level wavelet transform using Haar filter (Haar), 3-level wavelet transform using 3-tap Daubechies filter (Db3) and 3-level wavelet transform using 4-tap Daubechies filter (Db4).

1. ***Haar***
   The Haar wavelet employs a low-pass filter h^f_{Haar} and a high-pass filter g^f_{Haar}.
   where

$$h^f_{Haar} = [\sqrt{2}, \sqrt{2}] \tag{B.62}$$

$$g^f_{Haar} = [-\sqrt{2}, \sqrt{2}] \tag{B.63}$$

2. ***Db3***
   The Db3 wavelet employs a low-pass filter h^f_{Db3} and a high-pass filter g^f_{Db3}.
   where

$$h^f_{Db3} = [0.0352, -0.0854, -0.1350, 0.4598, 0.8068, 0.3326] \tag{B.64}$$

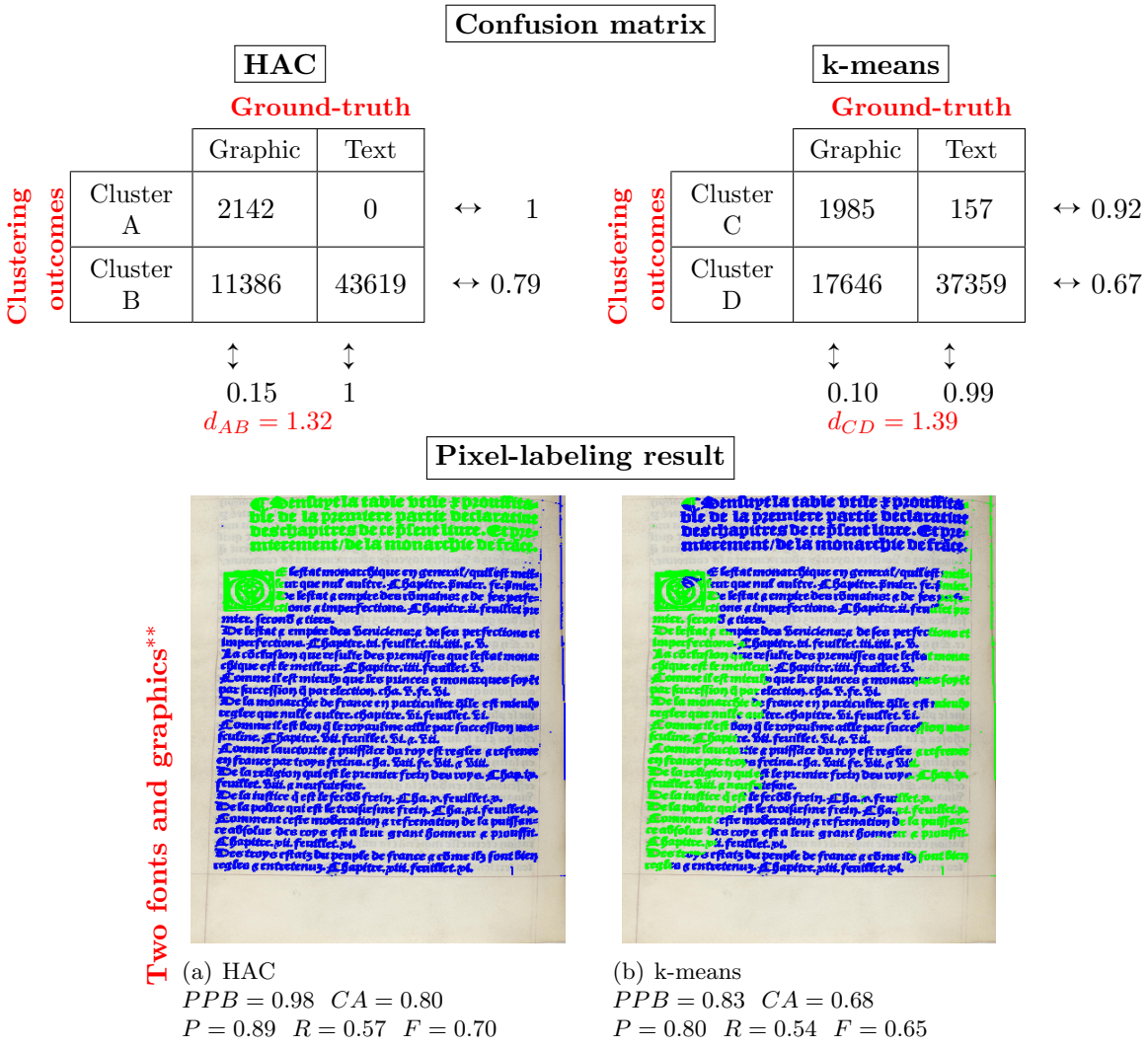
$$g^f_{Db3} = [-0.3326, 0.8068, -0.4598, -0.1350, 0.0854, 0.0352] \tag{B.65}$$

3. ***Db4***
   The Db4 wavelet employs a low-pass filter h^f_{Db4} and a high-pass filter g^f_{Db4}.
   where

$$h^f_{Db4} = [-0.0105, 0.0328, 0.0308, -0.1870, -0.0279, 0.6308, 0.7148, 0.2303] \tag{B.66}$$

$$g^f_{Db4} = [-0.2303, 0.7148, -0.6308, -0.0279, 0.1870, 0.0308, -0.0328, -0.0105] \tag{B.67}$$

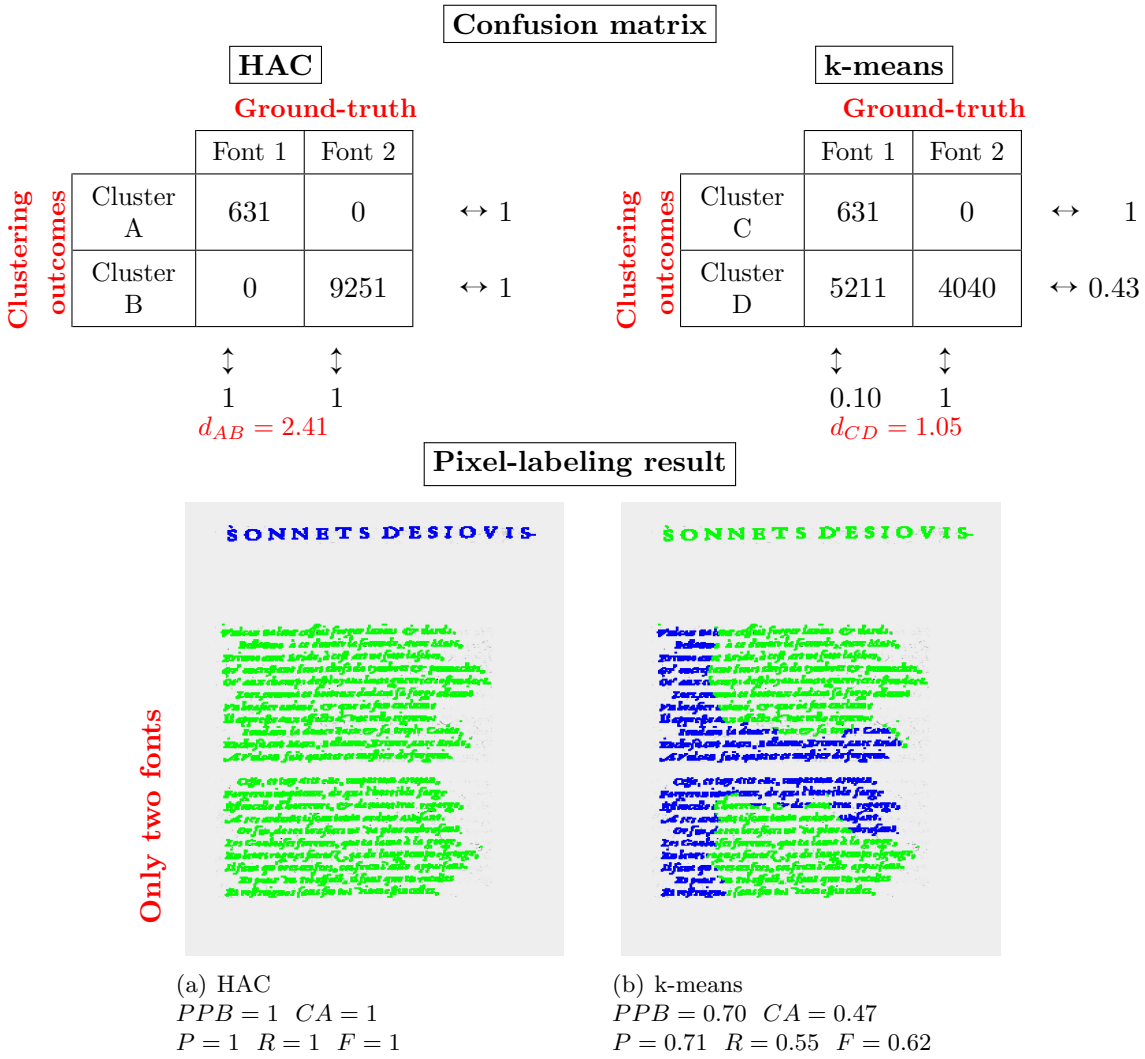
B.2. Visual results of using HAC vs. k-means in the pixel-clustering task of the proposed Gabor-based pixel-labeling scheme on the "DIGIDOC-Texture dataset"



(a) HAC
$PPB = 0.98$ $CA = 0.80$
$P = 0.89$ $R = 0.57$ $F = 0.70$

(b) k-means
$PPB = 0.83$ $CA = 0.68$
$P = 0.80$ $R = 0.54$ $F = 0.65$

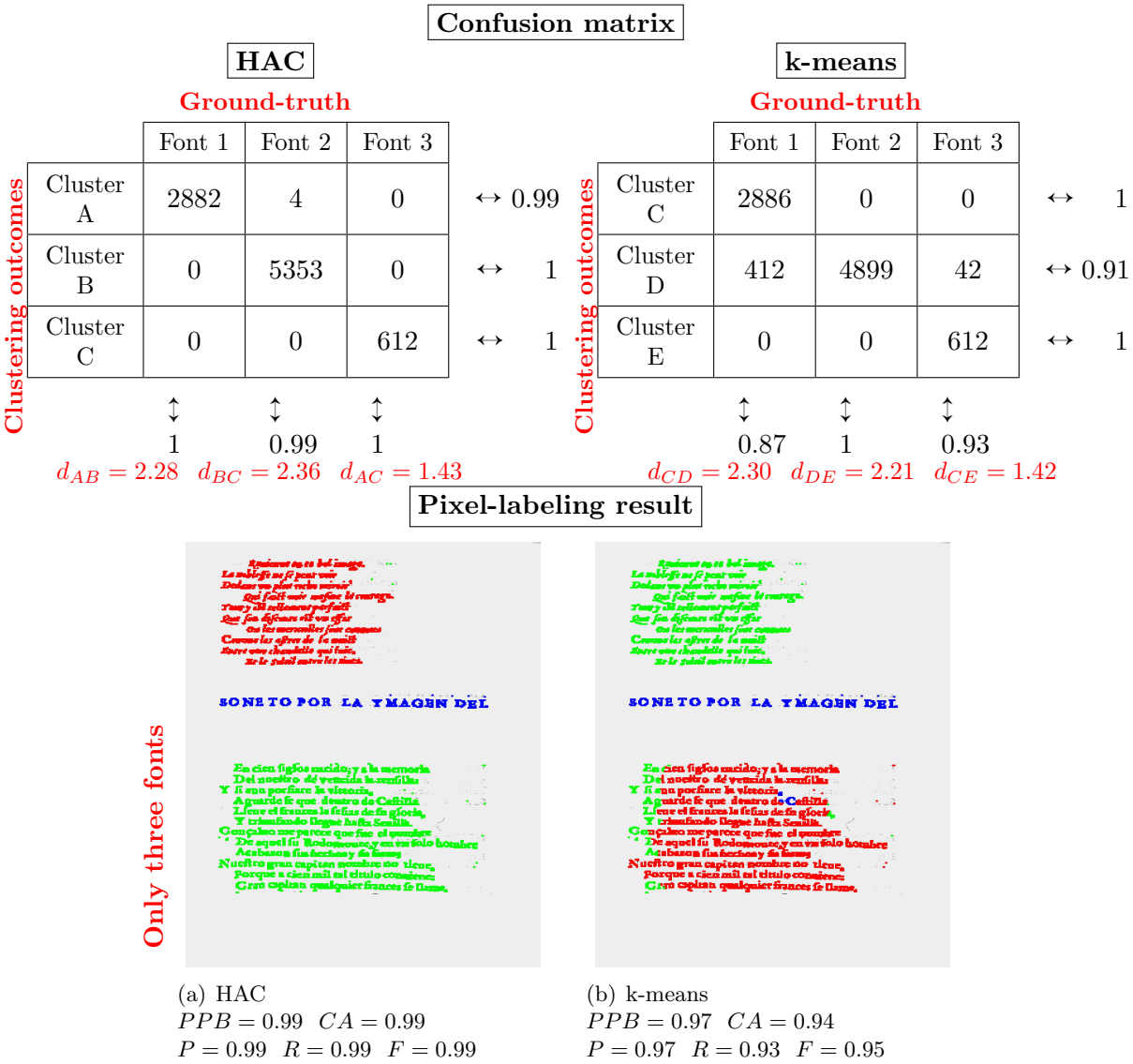
Figure B.23.: Examples of confusion matrix computation and pixel-labeling results of a document from the *"DIGIDOC-Texture dataset"*, containing graphics and two different text fonts ***"Two fonts and graphics**"***, obtained using the HAC and k-means algorithms, and by setting the maximum number of clusters to 2. Figure (a) represents the pixel-labeling result of a document containing graphics (green) and two different text fonts (blue and red) using the HAC algorithm. Figure (b) the pixel-labeling result of a document containing graphics (blue) and two different text fonts (green and red) using the k-means algorithm.



(a) HAC
$PPB = 1$ $CA = 1$
$P = 1$ $R = 1$ $F = 1$

(b) k-means
$PPB = 0.70$ $CA = 0.47$
$P = 0.71$ $R = 0.55$ $F = 0.62$

Figure B.24.: Examples of confusion matrix computation and pixel-labeling results of a document from the *"DIGIDOC-Texture dataset"*, containing text with two different fonts ***"Only two fonts"***, obtained using the HAC and k-means algorithms, and by setting the maximum number of clusters to 2. Figure (a) represents the pixel-labeling result of a document containing text with two different fonts, uppercase (blue) and italic (green) using the HAC algorithm. Figure (b) the pixel-labeling result of a document containing text with two different fonts, uppercase (green) and italic (blue) using the k-means algorithm.



(a) HAC
$PPB = 0.99$ $CA = 0.99$
$P = 0.99$ $R = 0.99$ $F = 0.99$

(b) k-means
$PPB = 0.97$ $CA = 0.94$
$P = 0.97$ $R = 0.93$ $F = 0.95$

Figure B.25.: Examples of confusion matrix computation and pixel-labeling results of a document from the *"DIGIDOC-Texture dataset"*, containing text with three different fonts ***"Only three fonts"***, obtained using the HAC and k-means algorithms, and by setting the maximum number of clusters to 3. Figure (a) represents the pixel-labeling result of a document containing text three two different fonts, normal (green), uppercase (blue) and italic (red) using the HAC algorithm. Figure (b) the pixel-labeling result of a document containing text with three different fonts, normal (red), uppercase (blue) and italic (gree) using the k-means algorithm.

B.3. Visual results of introducing vs. not introducing the "Pixel-labeling refinement" step into the auto-correlation and Gabor-based pixel-labeling scheme, illustrated in the "DIGIDOC-Texture dataset"

Auto-correlation

"Pixel-labeling"

(a) $PPB = 0.94$ $CA = 0.82$
$P = 0.63$ $R = 0.90$ $F = 0.74$

+ *"Pixel-labeling refinement"*

(b) $PPB = 0.95$ $CA = 0.81$
$P = 0.60$ $R = 0.90$ $F = 0.72$

Gabor

"Pixel-labeling"

(c) $PPB = 0.97$ $CA = 0.97$
$P = 0.94$ $R = 0.98$ $F = 0.96$

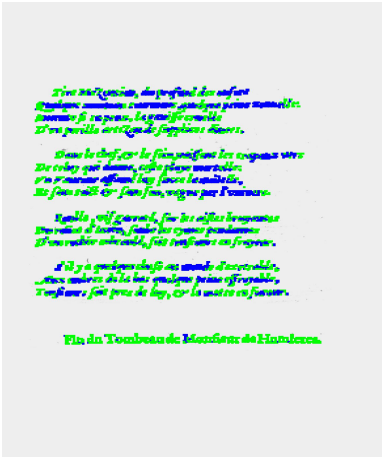
+ *"Pixel-labeling refinement"*

(d) $PPB = 0.94$ $CA = 0.97$
$P = 0.94$ $R = 0.98$ $F = 0.96$

Figure B.26.: Examples of introducing the ***"Pixel-labeling refinement"*** step into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics**"*** HDI from the *"DIGIDOC-Texture dataset"*.

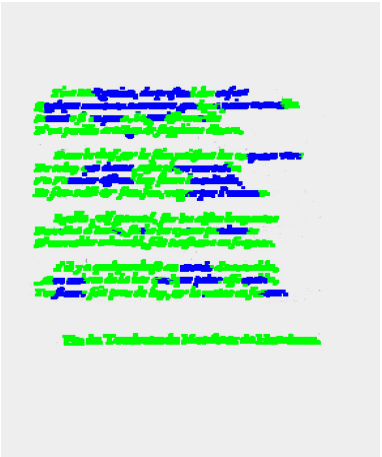
Auto-correlation

"Pixel-labeling"



(a) $PPB = 0.66$ $CA = 0.47$
$P = 0.69$ $R = 0.56$ $F = 0.62$

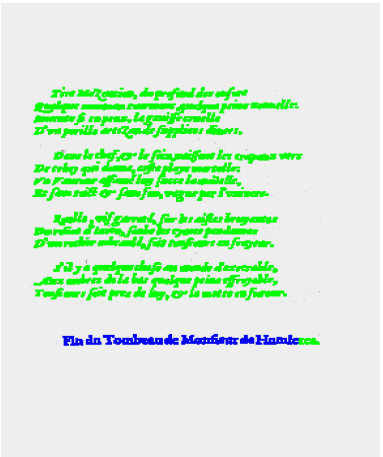
\+ *"Pixel-labeling refinement"*



(b) $PPB = 0.77$ $CA = 0.36$
$P = 0.65$ $R = 0.56$ $F = 0.60$

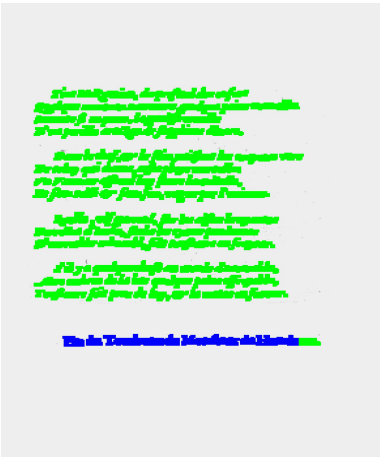
Gabor

"Pixel-labeling"



(c) $PPB = 0.99$ $CA = 0.99$
$P = 0.96$ $R = 0.99$ $F = 0.98$

\+ *"Pixel-labeling refinement"*

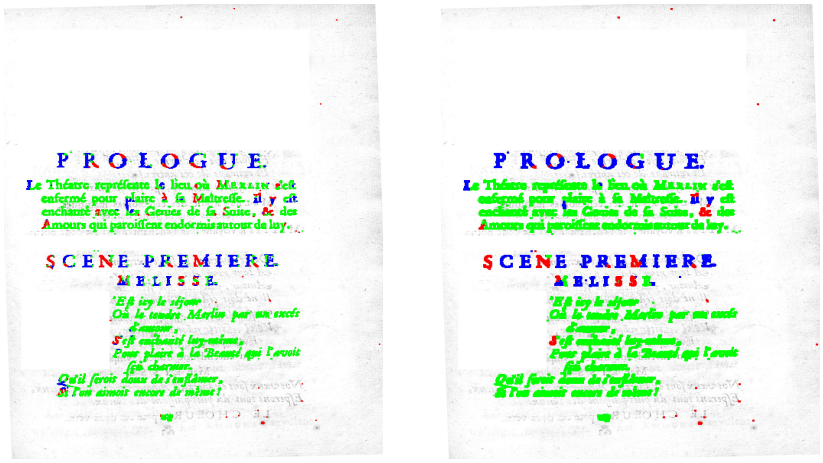


(d) $PPB = 0.99$ $CA = 0.99$
$P = 0.96$ $R = 0.99$ $F = 0.98$

Figure B.27.: Examples of introducing the ***"Pixel-labeling refinement"*** step into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only two fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.

Auto-correlation

"Pixel-labeling" | + "Pixel-labeling refinement"

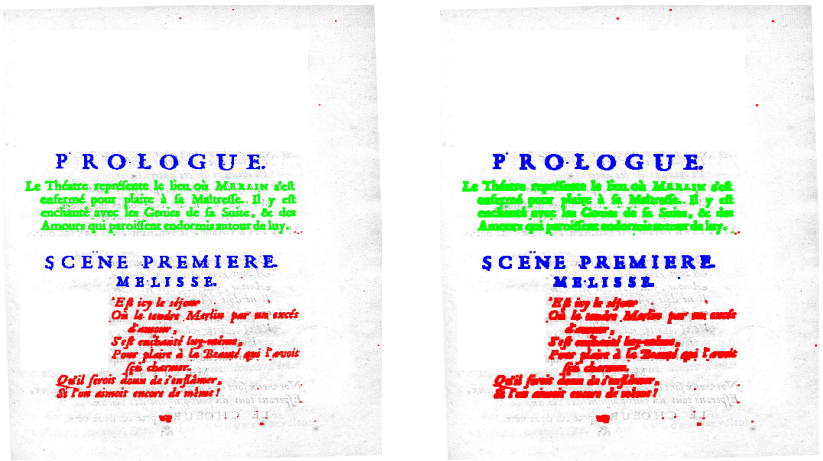


(a) $PPB = 0.78$ $CA = 0.57$ $P = 0.57$ $R = 0.57$ $F = 0.57$

(b) $PPB = 0.85$ $CA = 0.59$ $P = 0.60$ $R = 0.55$ $F = 0.58$

Gabor

"Pixel-labeling" | + "Pixel-labeling refinement"



(c) $PPB = 0.99$ $CA = 0.99$ $P = 0.99$ $R = 0.99$ $F = 0.99$

(d) $PPB = 0.99$ $CA = 0.99$ $P = 0.99$ $R = 0.99$ $F = 0.99$

Figure B.28.: Examples of introducing the ***"Pixel-labeling refinement"*** step into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only three fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.

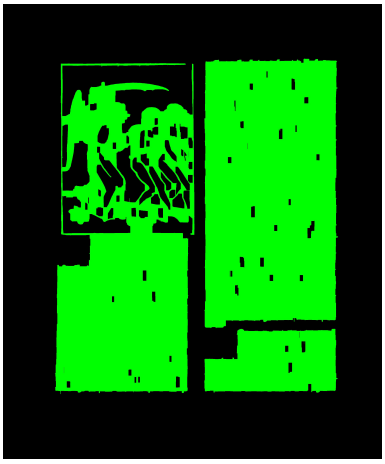
B.4. Visual results of introducing vs. not introducing the "Post-processing" step after the "Pixel-labeling refinement" task, into the auto-correlation and Gabor-based pixel-labeling scheme, illustrated in the "DIGIDOC-Texture dataset"

Auto-correlation

+ *"Pixel-labeling refinement"*

(a) $PPB = 0.95$ $CA = 0.81$
$P = 0.60$ $R = 0.90$ $F = 0.72$

+ *"Post-processing"*



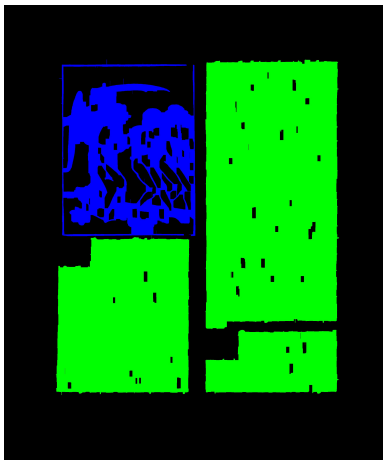
(b) $PPB = 1.00$ $CA = 0.76$
$P = 0.50$ $R = 0.38$ $F = 0.43$

Gabor

+ *"Pixel-labeling refinement"*

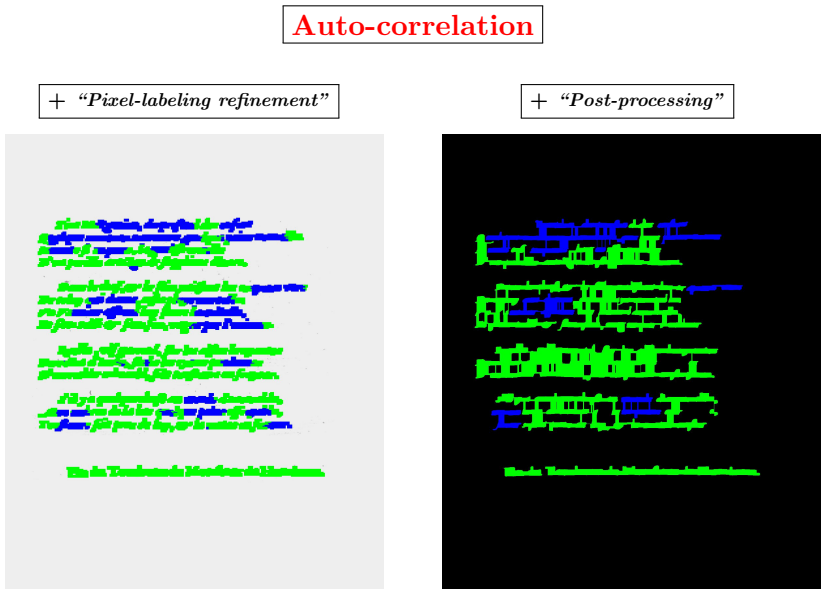
(c) $PPB = 0.94$ $CA = 0.97$
$P = 0.94$ $R = 0.98$ $F = 0.96$

+ *"Post-processing"*



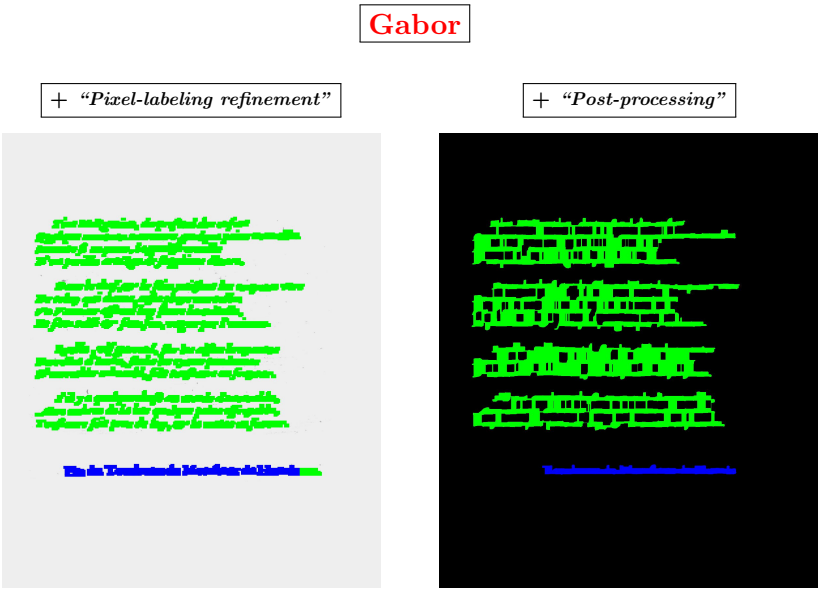
(d) $PPB = 1.00$ $CA = 1.00$
$P = 1.00$ $R = 1.00$ $F = 1.00$

Figure B.29.: Examples of introducing the ***"Post-processing"*** step after the *"Pixel-labeling refinement"* task, into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics**"*** HDI from the *"DIGIDOC-Texture dataset"*.



(a) $PPB = 0.77$ $CA = 0.36$
$P = 0.65$ $R = 0.56$ $F = 0.60$

(b) $PPB = 0.82$ $CA = 0.31$
$P = 0.62$ $R = 0.56$ $F = 0.59$



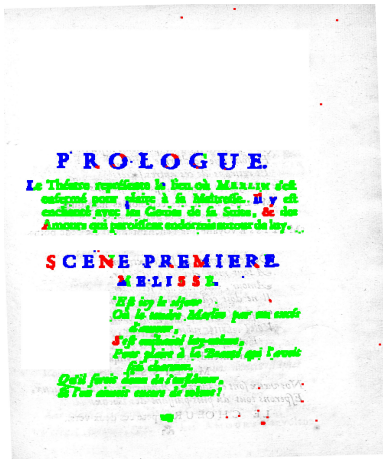
(c) $PPB = 0.99$ $CA = 0.99$
$P = 0.96$ $R = 0.99$ $F = 0.98$

(d) $PPB = 1.00$ $CA = 1.00$
$P = 1.00$ $R = 1.00$ $F = 1.00$

Figure B.30.: Examples of introducing the ***"Post-processing"*** step after the *"Pixel-labeling refinement"* task, into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only two fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.

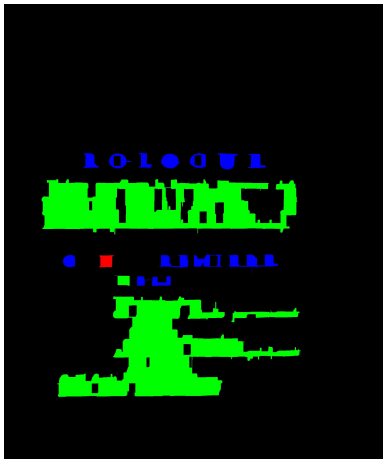
Auto-correlation

+ *"Pixel-labeling refinement"*



(a) $PPB = 0.85$ $CA = 0.59$
$P = 0.60$ $R = 0.55$ $F = 0.58$

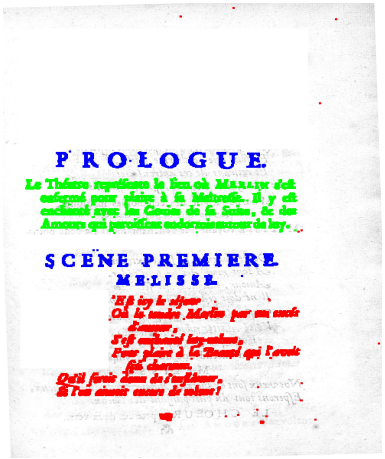
+ *"Post-processing"*



(b) $PPB = 0.91$ $CA = 0.63$
$P = 0.51$ $R = 0.57$ $F = 0.60$

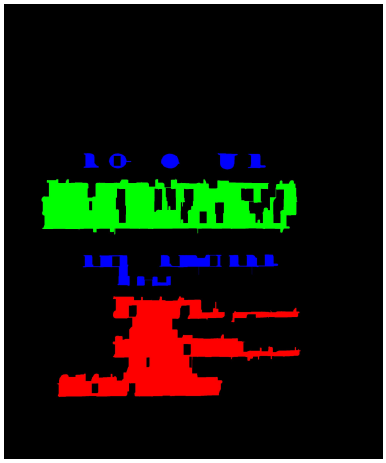
Gabor

+ *"Pixel-labeling refinement"*



(c) $PPB = 0.99$ $CA = 0.99$
$P = 0.99$ $R = 0.99$ $F = 0.99$

+ *"Post-processing"*



(d) $PPB = 1.00$ $CA = 1.00$
$P = 1.00$ $R = 1.00$ $F = 1.00$

Figure B.31.: Examples of introducing the ***"Post-processing"*** step after the *"Pixel-labeling refinement"* task, into the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only three fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.

B.5. Visual results of the "Homogeneous region extraction" step, performed after the "Post-processing" task on the auto-correlation and Gabor-based pixel-labeling scheme, illustrated in the "DIGIDOC-Texture dataset"

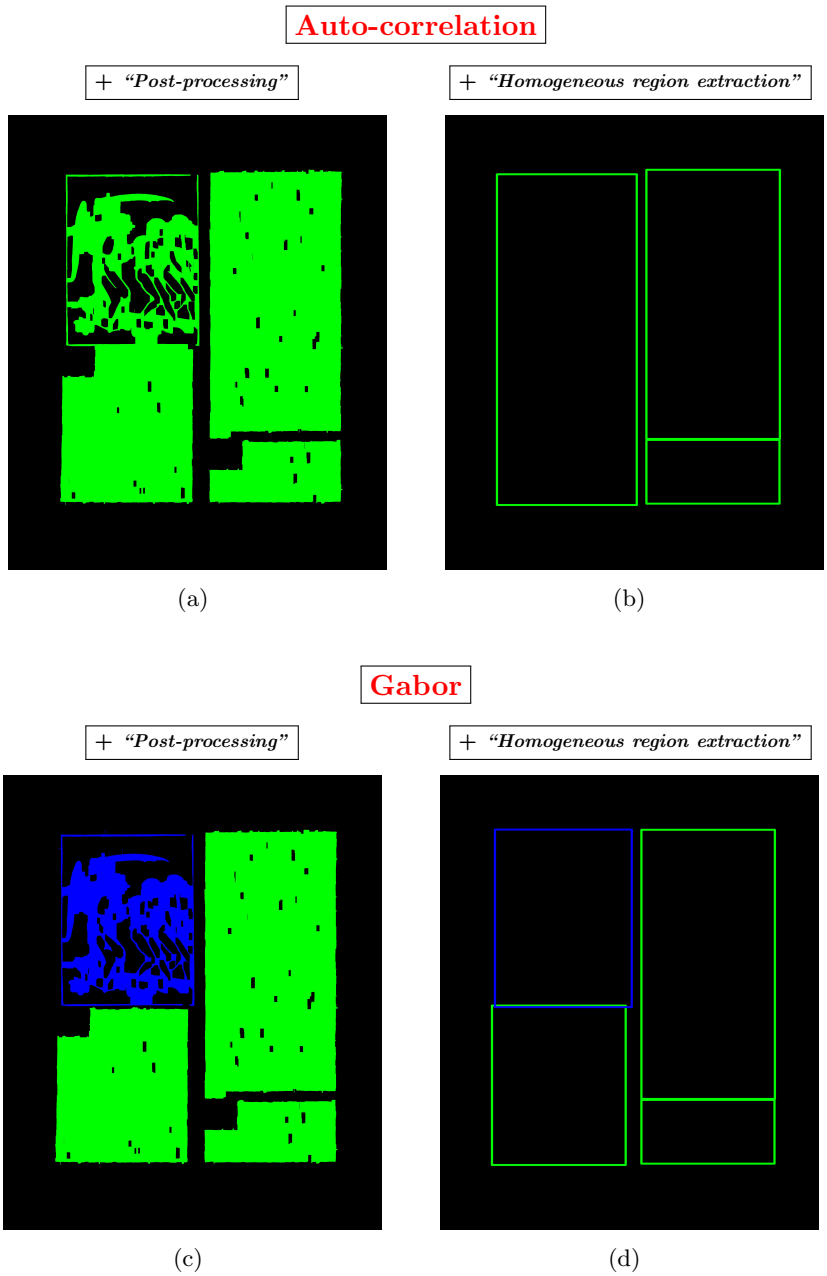


Figure B.32.: Examples of visual results of the ***"Homogeneous region extraction"*** step, performed after the *"Post-processing"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics**"*** HDI from the *"DIGIDOC-Texture dataset"*.

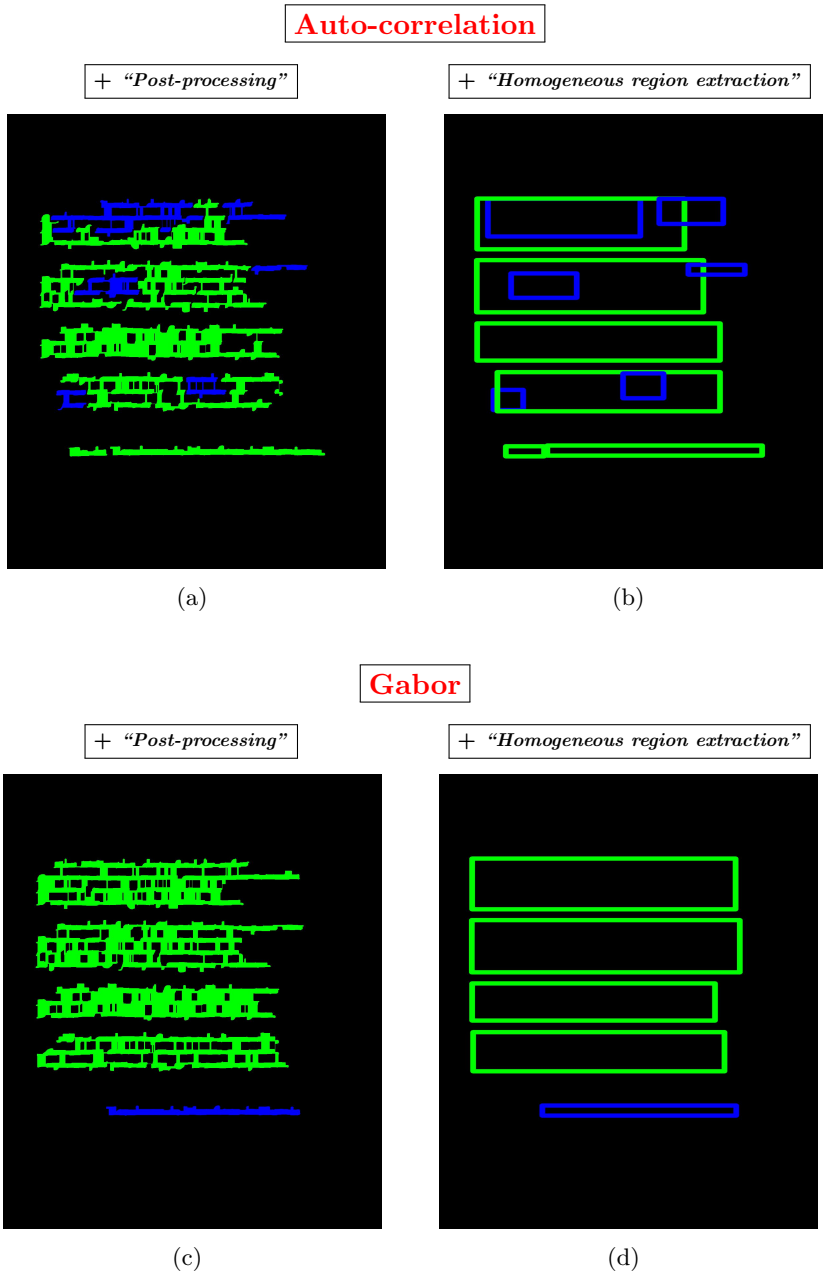


Figure B.33.: Examples of visual results of the ***"Homogeneous region extraction"*** task, performed after the *"Post-processing"* step on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only two fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.

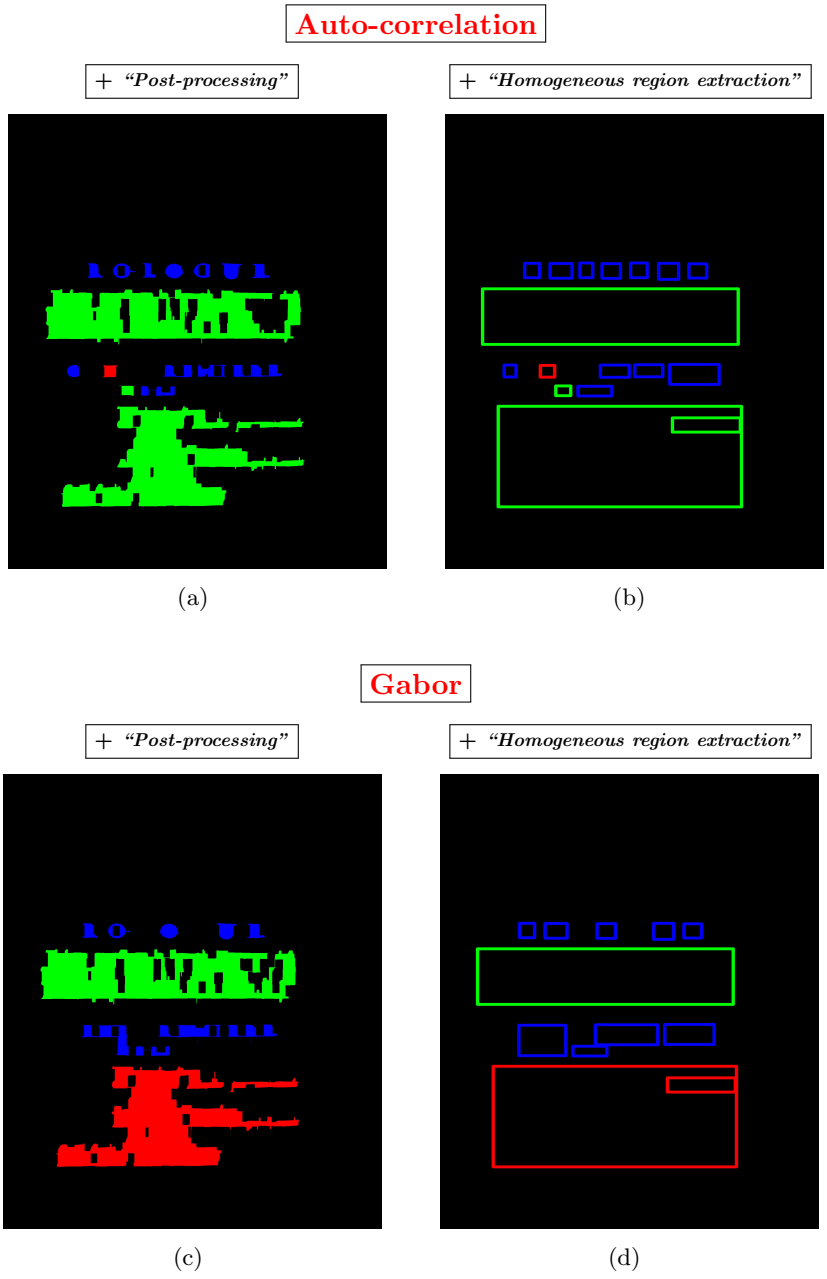
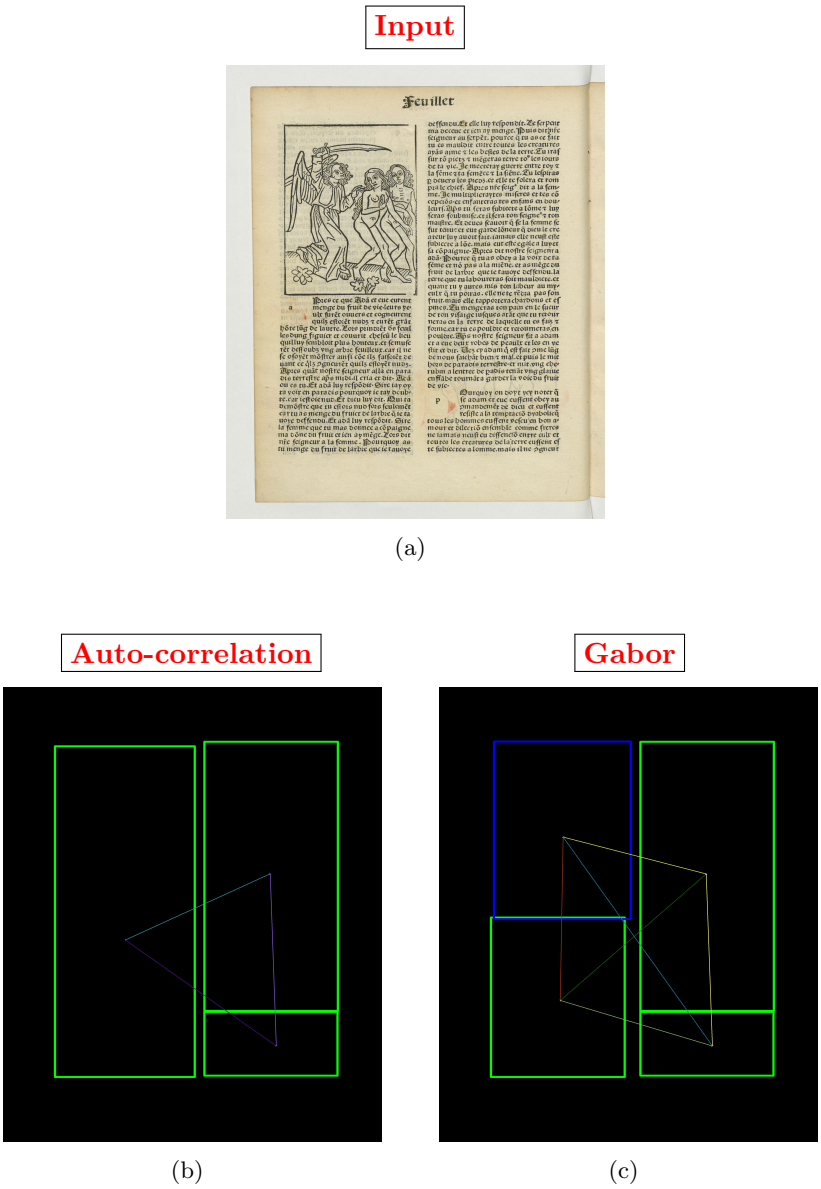


Figure B.34.: Examples of visual results of the ***"Homogeneous region extraction"*** task, performed after the *"Post-processing"* step on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only three fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.

B.6. Visual results of the "Structural signature generation" step, performed after the "Homogeneous region extraction" task on the auto-correlation and Gabor-based pixel-labeling scheme, illustrated in the "DIGIDOC-Texture dataset"

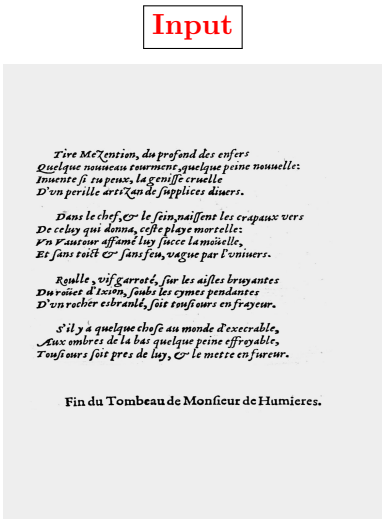


(a)

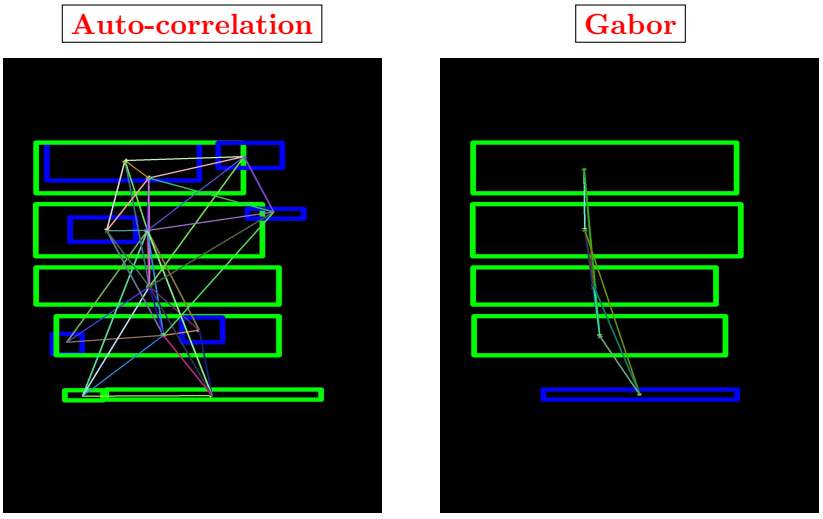
(b)

(c)

Figure B.35.: Examples of visual results of the **Structural signature generation** step, performed after the *"Homogeneous region extraction"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in a ***"Two fonts and graphics**"*** HDI from the *"DIGIDOC-Texture dataset"*.

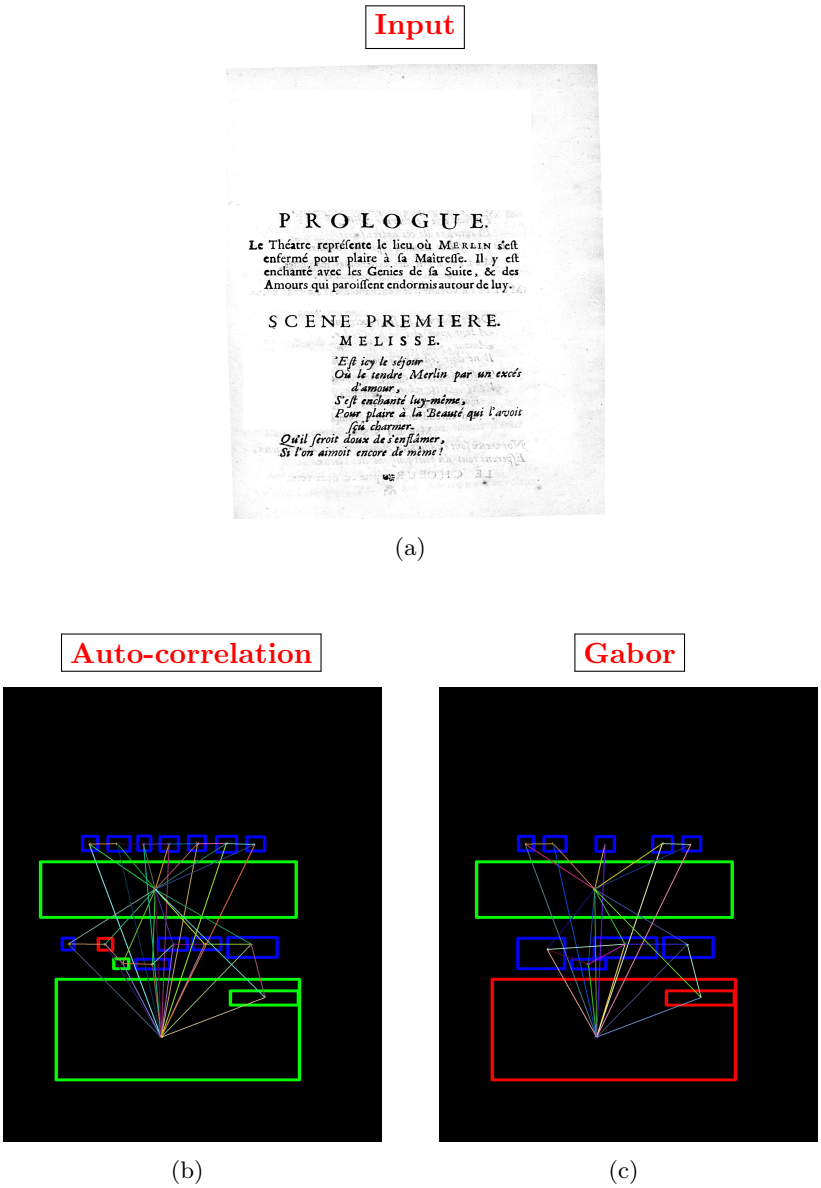


(a)



(b) (c)

Figure B.36.: Examples of visual results of the **Structural signature generation** step, performed after the *"Homogeneous region extraction"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only two fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.



(a)

(b)

(c)

Figure B.37.: Examples of visual results of the **Structural signature generation** step, performed after the *"Homogeneous region extraction"* task on the **auto-correlation** and **Gabor**-based pixel-labeling scheme, illustrated in an ***"Only three fonts"*** HDI from the *"DIGIDOC-Texture dataset"*.

B.7. A summary of the used moment attributes in this work

Among the computed vertex attributes, several kinds of moments are calculated. The most commonly moments are the regular (central and normalized central) and Hu moments which have been proposed as features to characterize patterns in classification and recognition applications [468]. Ten spatial moments (m_{ji}), seven central moments (μ_{ji}), seven normalized central moments (ν_{ji}) and seven Hu moments (hu_k) are computed to characterize the shape of the extracted homogeneous regions.

B.7.1. Spatial moments

The ten spatial moments (m_{ji}) which correspond to the $A^v_{16\rightarrow25}$ vertex attributes (m_{00}, m_{10}, m_{01}, m_{20}, m_{11}, m_{02}, m_{30}, m_{21}, m_{12} and m_{03}), are computed as:

$$m_{ji} = \sum_x \sum_y I(x,y) x^j y^i \tag{B.68}$$

B.7.2. Central moments

The seven central moments (μ_{ji}) which correspond to the $A^v_{26\rightarrow32}$ vertex attributes (μ_{20}, μ_{11}, μ_{02}, μ_{30}, μ_{21}, μ_{12} and μ_{03}), are computed as:

$$\mu_{ji} = \sum_x \sum_y I(x,y)(x-\bar{x})^j (y-\bar{y})^i \tag{B.69}$$

where $(\bar{x}, \bar{y})$ is the mass center:

$$\bar{x} = \frac{m_{10}}{m_{00}} \qquad \text{and} \qquad \bar{y} = \frac{m_{01}}{m_{00}} \tag{B.70}$$

B.7.3. Normalized central moments

The seven normalized central moments (ν_{ji}) which correspond to the $A^v_{33\rightarrow39}$ vertex attributes (ν_{20}, ν_{11}, ν_{02}, ν_{30}, ν_{21}, ν_{12} and ν_{03}), are computed as:

$$\nu_{ji} = \frac{\mu_{ji}}{m_{00}^{((i+j)/2)+1}} \tag{B.71}$$

B.7.4. Hu moments

The seven Hu moments (hu_k), where $k \in [0,6]$ (introduced by Hu [600]) which correspond to the $A^v_{40\rightarrow46}$ vertex attributes (hu_0, hu_1, hu_2, hu_3, hu_4, hu_5 and hu_6), are computed as:

$$\begin{cases} hu_0 = \nu_{20} + \nu_{02} \\ hu_1 = (\nu_{20} - \nu_{02})^2 + 4\nu_{11}^2 \\ hu_2 = (\nu_{30} - 3\nu_{12})^2 + (3\nu_{21} - \nu_{03})^2 \\ hu_3 = (\nu_{30} + \nu_{12})^2 + (\nu_{21} + \nu_{03})^2 \\ hu_4 = (\nu_{30} - 3\nu_{12})(\nu_{30} + \nu_{12})[(\nu_{30} + \nu_{12})^2 - 3(\nu_{21} + \nu_{03})^2] \\ \qquad + (3\nu_{21} - \nu_{03})(\nu_{21} + \nu_{03})[3(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] \\ hu_5 = (\nu_{20} - \nu_{02})[(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] + 4\nu_{11}(\nu_{30} + \nu_{12})(\nu_{21} + \nu_{03}) \\ hu_6 = (3\nu_{21} - \nu_{03})(\nu_{21} + \nu_{03})[3(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] \\ \qquad - (\nu_{30} - 3\nu_{12})(\nu_{21} + \nu_{03})[3(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] \end{cases} \tag{B.72}$$

B.8. Introduction to graphs and basic concepts

This section introduces a brief review of the basic definitions and concepts related to graphs.

Graph

A ***graph*** (G) is a well-known formalism of a structural representation in pattern recognition. It is composed of a finite set of vertices or nodes, connected by a set of edges (*cf.* Figure 6.2(b)). Vertices or nodes (G_v) represent distinct simple entities composing a complex pattern under consideration. Edges (G_e) represent the relationships between each two entities or parts of the analyzed pattern, where each edge connects two nodes in the graph G (*i.e.* $G_e = (G_v^s, G_v^d)$, such that both G_v^s and G_v^d are two vertices that belong to the set G_v). The graph size $|G|$ refers to the number of vertices (G_v) in the graph G.

- $G = (G_v, G_e)$ is a graph.
- G_v is a set of vertices or nodes.
- G_e is a set of edges that $G_e \subseteq G_v \times G_v$.

Simple graph

A graph G is said to be a ***simple*** when G is without loops (*i.e.* an edge that connects a vertex to itself) or multi-edges (*i.e.* more than one edge connecting two vertices).

Multi-graph

A graph G is said to be a ***multi-graph*** when G has several edges that may connect the same two vertices.

Directed graph

A graph G is said to be a ***directed*** when a direction is assigned to each edge of the set G_e. In fact, the edges $G_e^1 = (G_v^s, G_v^d)$ and $G_e^2 = (G_v^d, G_v^s)$ are different. Otherwise, G is said to be a ***undirected***.

Graph isomorphism

Two graphs $G = (G_v, G_e)$ and $G' = (G'_v, G'_e)$ are isomorphic (*i.e.* $G \simeq G'$), if and only there exists a bijective mapping $f : G_v \to G'_v$, where

$$(G_v, G'_v) \in G_e \Leftrightarrow (f(G_v), f(G'_v)) \in G'_e$$

Attributed graph

A graph G is said to be a ***attributed*** when:

- G is a four-tuple $G = (G_v, G_e, G_\mu, G_\nu)$.
- $G_\mu : G_v \rightarrow A^v$ is the vertex labeling function which associates the attribute or label a^v to a vertex G_v^i.
- $G_\nu : G_e \rightarrow A^e$ is the edge labeling function which associates the attribute or label a^e to a vertex G_e^i.
- A^v denotes a finite or infinite attribute or label set for G_v.
- A^e denotes a finite or infinite attribute or label set for G_e.
- A^v and/or A^e can be either continuous ($\in \Re$), discrete value or any combination of numeric and symbolic values.

Graph edit distance

The ***graph edit distance*** is a function $d(.,.)$ that:

$$\begin{array}{rcccl} d & : & \mathscr{G} \times \mathscr{G} & \rightarrow & \Re^+ \\ & & (G^1, G^2) & \mapsto & d(G^1, G^2) = \min\limits_{o=(o_1,\ldots,o_k)\in\Gamma(G^1,G^2)} \sum\limits_{i=1}^{k} c(o_i) \end{array}$$

where

- $G^1 = (G_v^1, G_e^1, G_\mu^1, G_\nu^1)$ and $G^2 = (G_v^2, G_e^2, G_\mu^2, G_\nu^2)$ are two graphs from the set $\mathscr{G}$.
- $\Gamma(G^1, G^2)$ is the set of all edit operations $o = (o_1, \ldots, o_k)$, allowing to transform G^1 into G^2.
- $c(.)$ is a cost function on an elementary edit operation o_i.

Elementary edit operation

An ***elementary edit operation*** o_i is one of:

- Vertex substitution: $v^1 \rightarrow v^2$
- Edge substitution: $e^1 \rightarrow e^2$
- Vertex deletion: $v^1 \rightarrow \epsilon$
- Edge deletion: $e^1 \rightarrow \epsilon$
- Vertex insertion: $\epsilon \rightarrow v^2$
- Edge insertion: $\epsilon \rightarrow e^2$

where

- $v^1 \in G_v^1$, $v^2 \in G_v^2$, $e^1 \in G_e^1$ and $e^2 \in G_e^2$.
- ϵ is a dummy vertex or edge which is used to model insertion or deletion operations.

Cost function

A ***cost function*** $c(.)$ on an elementary edit operation o_i must satisfy the following criteria:

- $c(v^1 \rightarrow v^2) \leq c(v^1 \rightarrow v) + c(v \rightarrow v^2)$
- $c(e^1 \rightarrow e^2) \leq c(e^1 \rightarrow e) + c(e \rightarrow e^2)$
- $c(v^1 \rightarrow \epsilon) \leq c(v^1 \rightarrow v) + c(v \rightarrow \epsilon)$
- $c(e^1 \rightarrow \epsilon) \leq c(e^1 \rightarrow e) + c(e \rightarrow \epsilon)$
- $c(\epsilon \rightarrow v^2) \leq c(\epsilon \rightarrow v) + c(v \rightarrow v^2)$
- $c(\epsilon \rightarrow e^2) \leq c(\epsilon \rightarrow e) + c(e \rightarrow e^2)$

Moreover, the cost functions have to be defined in a symmetric manner to guarantee the symmetry property of the graph edit distance (*i.e.* $d(G^1, G^2) = d(G^2, G^1)$). Indeed, the following criteria have to be satisfied by checking the same cost for the reverse edit path:

- $c(v^1 \rightarrow v^2) = c(v^2 \rightarrow v^1)$
- $c(e^1 \rightarrow e^2) = c(e^2 \rightarrow e^1)$
- $c(v^1 \rightarrow \epsilon) = c(\epsilon \rightarrow v^1)$
- $c(e^1 \rightarrow \epsilon) = c(\epsilon \rightarrow e^1)$

Only the paths corresponding to matchings between the compared graphs provided that all vertices (resp. edges) for each graph are either matched to a vertex (resp. edge) from the other graph (*i.e.* substitution or one-to-one mapping) or matched to a dummy vertex (resp. edge) (*i.e.* deletion/zero-to-one mapping or insertion/one-to-zero mapping), are selected.

These mappings define the graph edit distance by computing the minimum value among the costs associated to edit paths. Topological constraints must be respected when computing the graph edit distance for inexact graph-matching that if two edges are matched, their end vertices have to be matched also.

Graph edit distance between unlabeled graphs

A ***graph edit distance between unlabeled graphs*** is computed based on the identity property (*i.e.* $d(G^1, G^2) = 0 \Rightarrow G^1 = G^2$). Indeed, the following statements can be deduced:

- The substitution costs are equal to zero.
- The insertion and deletion costs are set to a constant.

The minimum cost edit path to transform G^1 into G^2 is computed based on the edit operations defined in $D^1 \cup D^2$ that any vertex deletion is preceded by the deletion of connected edges, and that any edge insertion is preceded by the insertion of end vertices. where

- D^1 is a set of edit operations that are required to transform G^1 to $\hat{G}$.
- D^2 is a set of edit operations that are required to transform $\hat{G}$ to G^2.
- $\hat{G}$ is a maximum common sub-graph of G^1 and G^2.

If all edit operations from D^2 are first applied, G^1 is first transformed into $\check{G}$. Then, edit operations from D^1 transform $\check{G}$ into G^2, where $\check{G}$ denotes a minimum common super-graph of G^1 and G^2.

$\check{G}$ is a super-graph of G^1 and G^2 if G^1 and G^2 are both sub-graphs of $\check{G}$.

Graph edit distance between attributed graphs

A ***graph edit distance between attributed graphs*** is computed based on the edit costs which are generally defined as functions of vertices (resp. edges) labels.

- The substitution costs are defined as a function of the labels of the substituted vertices (resp. edges):

$$\begin{cases} c(v^1 \to v^2) = c(v^2 \to v^1) = f_v(G^1_\mu(v^1), G^2_\mu(v^2)) \\ c(e^1 \to e^2) = c(e^2 \to e^1) = f_e(G^1_\nu(e^1), G^2_\nu(e^2)) \end{cases}$$

  where f_v and f_e denote the substitution cost function of the labels of the substituted vertices and edges, respectively.

- The insertion/deletion costs are defined according to the label of the inserted/deleted vertex (resp. edge):

$$\begin{cases} c(v^1 \to \epsilon) = c(\epsilon \to v^1) = g_v(G_\mu) \\ c(e^1 \to \epsilon) = c(\epsilon \to e^1) = g_e(G_\nu) \end{cases}$$

  where g_v and g_e denote the insertion/deletion cost function of the labels of the inserted/deleted vertex and edge, respectively.

B.9. Graph edit distance using a binary linear programming

This section introduces a brief review of the used graph edit distance (GED) by means of a binary linear programming (BLP). In this work, a binary linear programming (BLP) is used to model the GED paradigm. An approximate GED approach is used based on a lower bound of the exact GED provided by the relaxation of the BLP formulation.

B.9.1. Binary linear programming

A binary linear programming (BLP) is a derivative of integer linear programming (ILP) where the variables are binary. A general form of a BLP is defined by the following optimization problem:

$$\begin{cases} \textbf{\textit{Objective function:}} & \min\limits_{x} c^{\intercal} x \\ \textbf{\textit{Linear constraint:}} & \text{subject to } Ax \leq b \\ \textbf{\textit{Domain constraint:}} & \text{with } x \in \mathbb{Z}^n \end{cases} \tag{B.73}$$

where $c \in \Re^n$, $A \in \Re^{n \times m}$ and $b \in \Re^m$ are data used to solve the optimization problem. A solution of this optimization problem is a vector x of n binary variables. If this optimization problem has admissible solutions, the optimal solution is the one that minimizes the objective function and respects the two constraints defined (B.73).

B.9.2. Modeling graph edit distance with binary linear programming

Since our goal is compute the GED between two attributed directed graphs $G^1 = (G^1_v, G^1_e, G^1_\mu, G^1_\nu)$ and $G^2 = (G^2_v, G^2_e, G^2_\mu, G^2_\nu)$ (*cf.* Chapter B and particularly Section B.8), in this work GED paradigm is formulated as a BLP. The formulations in this section is given for simple directed graphs. Nevertheless, these formulations can also be applied relatively simply to multi-graph and/or undirected graphs. In Appendix B and particularly in Section B.8, a detailed description of the GED and its three types of elementary edit operation used to match the two graphs G^1 and G^2:

1. The ***substitution*** of the label of a vertex (resp. an edge) of G^1 with the label of a vertex (resp. an edge) of G^2,
2. The ***deletion*** of a vertex (resp. an edge) from G^1,
3. The ***insertion*** of a vertex (resp. an edge) of G^2 in G^1.

For each type of elementary edit operation, a set of binary variables which is used to define an edit path between the graphs G^1 and G^2 by means of a 6-tuple (x, y, u, v, e, f). Table B.1 presents the defined set of binary variables for each type of edit operation corresponding to a BLP used to model the GED paradigm. Then, cost functions which depend on the labels of vertices and edges, are defined for each type of elementary edit operation in order to evaluate the cost of an edit path. Table B.2 presents the defined cost functions for each type of elementary edit operation.

Table B.1.: A set of binary variables for each type of edit operation corresponding to a BLP used to model the GED paradigm.

	Type	**Id.**	**Binary variable**
Substitution	Vertex	x	$\forall (i,k) \in G^1_v \times G^2_v,\ x_{i,k} = \begin{cases} 1, \text{ if } i \text{ is substituted with } k, \\ 0, \text{ otherwise.} \end{cases}$
	Edge	y	$\forall (ij,kl) \in G^1_e \times G^2_e,\ y_{ij,kl} = \begin{cases} 1, \text{ if } ij \text{ is substituted with } kl, \\ 0, \text{ otherwise.} \end{cases}$

Table B.1 – continued from previous page

	Type	Id.	Binary variable
Deletion	Vertex	u	$\forall i \in G_v^1,\ u_i = \begin{cases} 1, \text{ if } i \text{ is deleted from } G_v^1, \\ 0, \text{ otherwise.} \end{cases}$
	Edge	e	$\forall ij \in G_e^1,\ e_{ij} = \begin{cases} 1, \text{ if } ij \text{ is deleted from } G_v^1, \\ 0, \text{ otherwise.} \end{cases}$
Insertion	Vertex	v	$\forall k \in G_v^2,\ v_k = \begin{cases} 1, \text{ if } k \text{ is inserted in } G_v^1, \\ 0, \text{ otherwise.} \end{cases}$
	Edge	f	$\forall kl \in G_e^2,\ f_{kl} = \begin{cases} 1, \text{ if } kl \text{ is inserted in } G_v^1, \\ 0, \text{ otherwise.} \end{cases}$

Table B.2.: The defined cost functions for each type of elementary edit operation corresponding to a BLP used to model the GED paradigm.

	Type	Id.	Binary variable	Model	Target	Cost	Description
Substitution	Vertex	x	$x_{i,k}$	i	k	$c(i \to k)$	$\forall (i,k) \in G_v^1 \times G_v^2$, substituting the vertex i with k
	Edge	y	$y_{ij,kl}$	ij	kl	$c(ij \to kl)$	$\forall (ij,kl) \in G_e^1 \times G_e^2$, substituting the edge ij with kl
Deletion	Vertex	u	u_i	i	$\varnothing$	$c(i \to \epsilon)$	$\forall i \in G_v^1$, deleting the vertex i from the graph G^1
	Edge	e	e_{ij}	ij	$\varnothing$	$c(ij \to \epsilon)$	$\forall ij \in G_e^1$, deleting the edge ij from the graph G^1
Insertion	Vertex	v	v_k	$\varnothing$	k	$c(\epsilon \to k)$	$\forall k \in G_v^2$, inserting the vertex k in the graph G^1
	Edge	f	f_{kl}	$\varnothing$	kl	$c(\epsilon \to kl)$	$\forall kl \in G_e^2$, inserting the edge kl in the graph G^1

Afterwards, to compute the GED between the graphs G^1 and G^2, an overall cost can be deduced by applying an edit path defined with the 6-tuple (x, y, u, v, e, f) on graph G^1 to make it isomorphic to the graph G^2. Nevertheless, this overall cost must be minimized by means of the following objective function:

$$\min_{x,y,u,v,e,f} \left(\begin{array}{l} \displaystyle\sum_{i \in G_v^1} \sum_{k \in G_v^2} c(i \to k)\ x_{i,k} + \sum_{ij \in G_e^1} \sum_{kl \in G_e^2} c(ij \to kl)\ y_{ij,kl} \\ \displaystyle + \sum_{i \in G_v^1} c(i \to \epsilon)\ u_i + \sum_{ij \in G_e^1} c(ij \to \epsilon)\ e_{ij} \\ \displaystyle + \sum_{k \in G_v^2} c(\epsilon \to k)\ v_k + \sum_{kl \in G_{kl}^2} c(\epsilon \to kl)\ f_{kl} \end{array} \right) \tag{B.74}$$

Subsequently, few linear constraints must be respected to have admissible edit path solutions of the BLP that minimizes the objective function applied on the 6-tuple (x, y, u, v, e, f) on the graph G^1 to make it isomorphic to the graph G^2 (*cf.* equation B.74). A solution is considered as admissible if and only if the following linear constraints related to the involved edit path solution are respected:

1. ***Vertex matching constraints:*** The edit path solution provides an one-to-one mapping (*i.e.* vertex substitution) between two sub-sets of the G^1 and G^2 vertices. The remaining vertices are either deleted or inserted (*i.e.* vertex deletion/insertion). As a consequence, two linear vertex matching constraints can be deduced: *(i)* each vertex of the graph G^1 is either matched to exactly one vertex of the graph G^2 or deleted from the graph G^1 (*cf.* equation B.75) and *(ii)* each vertex of the graph G^2 is either matched to exactly one vertex of the graph G^1 or inserted in the graph G^1 (*cf.* equation B.76).

2. ***Edge matching constraints:*** The edit path solution provides an one-to-one mapping (*i.e.* edge substitution) between two sub-sets of the G^1 and G^2 edges. The remaining edges are either deleted or inserted (*i.e.* edge deletion/insertion). Similarly to the deduced two linear vertex matching constraints, two linear vertex matching constraints can be defined: *(i)* each edge of the graph G^1 is either matched to exactly one edge of the graph G^2 or deleted from the graph G^1 (*cf.* equation B.77) and *(ii)* each edge of the graph G^2 is either matched to exactly one edge of the graph G^1 or inserted in the graph G^1 (*cf.* equation B.78).

3. ***Topological constraints:*** The edit path solution provides a consistent vertice and edge matchings (*i.e.* graph topology is respected). Indeed, the following statement can be deduced: an edge $ij \in G_e^1$ can be matched to an edge $kl \in G_e^2$ only if the source vertices $i \in G_v^1$ and $k \in G_v^2$ on the one hand, and the destination vertices $j \in G_v^1$ and $l \in G_v^2$ on the other hand, are respectively matched. As a matter of fact, two linear topological matching constraints can be defined: *(i)* ij and kl can be matched only if their source vertices are matched (*cf.* equation B.79) and *(ii)* ij and kl can be matched only if their destination vertices are matched (*cf.* equation B.80).

Table B.3.: A defined set of linear constraints to guarantee an admissible edit path solution corresponding to a BLP used to model the GED paradigm.

Constraint type	**Equation**	**Description**
Vertex matching	$\forall i \in G_v^1, \sum_{k \in G_v^2} x_{i,k} + u_i = 1$ (B.75)	Each vertex of the graph G^1 is either matched to exactly one vertex of the graph G^2 or deleted from the graph G^1.
	$\forall k \in G_v^2, \sum_{i \in G_v^1} x_{i,k} + v_k = 1$ (B.76)	Each vertex of the graph G^2 is either matched to exactly one vertex of the graph G^1 or inserted in the graph G^1.
Edge matching	$\forall ij \in G_e^1, \sum_{kl \in G_e^2} y_{ij,kl} + e_{ij} = 1$ (B.77)	Each edge of the graph G^1 is either matched to exactly one edge of the graph G^2 or deleted from the graph G^1.
	$\forall kl \in G_e^2, \sum_{ij \in G_e^1} y_{ij,kl} + f_{kl} = 1$ (B.78)	Each edge of the graph G^2 is either matched to exactly one edge of the graph G^1 or inserted in the graph G^1.

Table B.3 – continued from previous page

Constraint type	Equation	Description
Topological	$\forall(ij,kl) \in G_e^1 \times G_e^2,\ y_{ij,kl} \leq x_{i,k}$ (B.79)	ij and kl can be matched only if their source vertices are matched.
	$\forall(ij,kl) \in G_e^1 \times G_e^2,\ y_{ij,kl} \leq x_{j,l}$ (B.80)	ij and kl can be matched only if their destination vertices are matched.

Finally, the domain constraints are defined to ensure that the edit path solution is binary as follows:

$$\begin{cases} \forall(i,k) \in G_v^1 \times G_v^2, & x_{i,k} \in \{0,1\} \\ \forall(ij,kl) \in G_e^1 \times G_e^2, & y_{ij,kl} \in \{0,1\} \\ \forall i \in G_v^1, & u_i \in \{0,1\} \\ \forall k \in G_v^2, & v_k \in \{0,1\} \\ \forall ij \in G_e^1, & e_{ij} \in \{0,1\} \\ \forall kl \in G_e^2, & f_{kl} \in \{0,1\} \end{cases} \tag{B.81}$$

Therefore, based on the defined objective function (*cf.* equation B.74), domain (*cf.* equations B.81) and linear (*cf.* Table B.3) constraints, an admissible edit path solution of the BLP that minimizes the objective function applied on the graph G^1 to make it isomorphic to the graph G^2, is given based on the BLP formulation of GED which is illustrated in Table B.4. The BLP formulation of GED has:

- $|G_v^1|+|G_v^2|+|G_e^1|+|G_e^2|+|G_v^1||G_v^2|+|G_e^1||G_e^2|$ variables,
- $|G_v^1|+|G_v^2|+|G_e^1|+|G_e^2|+2|G_e^1||G_e^2|$ constraints without taking into consideration the domain constraints (*cf.* equations (B.81), (B.81), (B.81), (B.81), (B.81) and (B.81)).

Table B.4.: BLP formulation of the GED paradigm.

BLP id	Value
Objective function	$\min_{x,y,u,v,e,f} \left(\begin{array}{l} \sum\limits_{i \in G_v^1} \sum\limits_{k \in G_v^2} c(i \to k)\ x_{i,k} + \sum\limits_{ij \in G_e^1} \sum\limits_{kl \in G_e^2} c(ij \to kl)\ y_{ij,kl} \\ + \sum\limits_{i \in G_v^1} c(i \to \epsilon)\ u_i + \sum\limits_{ij \in G_e^1} c(ij \to \epsilon)\ e_{ij} \\ + \sum\limits_{k \in G_v^2} c(\epsilon \to k)\ v_k + \sum\limits_{kl \in G_{kl}^2} c(\epsilon \to kl)\ f_{kl} \end{array} \right)$
Linear constraints	$\forall i \in G_v^1,\ \sum\limits_{k \in G_v^2} x_{i,k} + u_i = 1$

Table B.4 – continued from previous page

BLP id	Value
	$\forall k \in G_v^2,\ \sum_{i \in G_v^1} x_{i,k} + v_k = 1$
	$\forall ij \in G_e^1,\ \sum_{kl \in G_e^2} y_{ij,kl} + e_{ij} = 1$
	$\forall kl \in G_e^2,\ \sum_{ij \in G_e^1} y_{ij,kl} + f_{kl} = 1$
	$\forall (ij,kl) \in G_e^1 \times G_e^2,\ y_{ij,kl} \leq x_{i,k}$
	$\forall (ij,kl) \in G_e^1 \times G_e^2,\ y_{ij,kl} \leq x_{j,l}$
Domain constraints	$\forall (i,k) \in G_v^1 \times G_v^2,\ x_{i,k} \in \{0,1\}$
	$\forall (ij,kl) \in G_e^1 \times G_e^2,\ y_{ij,kl} \in \{0,1\}$
	$\forall i \in G_v^1,\ u_i \in \{0,1\}$
	$\forall k \in G_v^2,\ v_k \in \{0,1\}$
	$\forall ij \in G_e^1,\ e_{ij} \in \{0,1\}$
	$\forall kl \in G_e^2,\ f_{kl} \in \{0,1\}$

B.9.3. Optimized binary linear programming formulation for modeling graph edit distance

In this section, an optimized BLP formulation for modeling GED with less variables and constraints than those used in Section B.9.2. This optimized BLP formulation depends on the size/density of the graphs into consideration. Based on the BLP formulation of GED (*cf.* Table B.4), the following

variables, u, v, e and e, are unnecessary and greatly increase the computational time. As a matter of fact, the following equations (B.75), (B.76), (B.77) and (B.78), are transformed into linear inequality constraints (*cf.* Table B.5). Then, the topological constraints (*cf.* equations (B.79) and (B.80)) which have been previously presented in Section B.9.2, can be expressed mathematically in a different way to have clear and precise information, without adversely affecting the binary edit path solutions (*cf.* equations (B.86) and (B.87)).

Table B.5.: A defined set of linear inequality constraints to guarantee an admissible edit path solution corresponding to an optimized BLP formulation used to model the GED paradigm.

Constraint type	**Equation**	**Description**
Vertex matching	$\forall i \in G_v^1, \sum_{k \in G_v^2} x_{i,k} \leq 1$ (B.82)	Each vertex of the graph G^1 is either matched to exactly one vertex of the graph G^2 or deleted from the graph G^1.
	$\forall k \in G_v^2, \sum_{i \in G_v^1} x_{i,k} \leq 1$ (B.83)	Each vertex of the graph G^2 is either matched to exactly one vertex of the graph G^1 or inserted in the graph G^1.
Edge matching	$\forall ij \in G_e^1, \sum_{kl \in G_e^2} y_{ij,kl} \leq 1$ (B.84)	Each edge of the graph G^1 is either matched to exactly one edge of the graph G^2 or deleted from the graph G^1.
	$\forall kl \in G_e^2, \sum_{ij \in G_e^1} y_{ij,kl} \leq 1$ (B.85)	Each edge of the graph G^2 is either matched to exactly one edge of the graph G^1 or inserted in the graph G^1.
Topological	$\forall k \in G_v^2 \forall ij \in G_e^1, \sum_{kl \in G_e^2} y_{ij,kl} \leq x_{i,k}$ (B.86)	Provided an edge $ij \in G_e^1$ and a vertex $k \in G_v^2$, there is at most one edge incident away from k that can be matched with ij.
	$\forall l \in G_v^2 \forall ij \in G_e^1, \sum_{kl \in G_e^2} y_{ij,kl} \leq x_{j,l}$ (B.87)	Provided an edge $ij \in G_e^1$ and a vertex $l \in G_v^2$, there is at most one edge incident towards l that can be matched with ij.

Adterwards, by replacing the u, v, e and e variables by their expressions which are deduced from the equations (B.75), (B.76), (B.77) and (B.78), respectively, in the objective function (B.74), a new formulation of an objective function is given by

$$
\begin{aligned}
&\min_{x,y,u,v,e,f} \left(\begin{array}{l}
\displaystyle\sum_{i \in G_v^1} \sum_{k \in G_v^2} c(i \to k)\ x_{i,k} + \sum_{ij \in G_e^1} \sum_{kl \in G_e^2} c(ij \to kl)\ y_{ij,kl} \\
\displaystyle + \sum_{i \in G_v^1} c(i \to \epsilon)\ u_i + \sum_{ij \in G_e^1} c(ij \to \epsilon)\ e_{ij} \\
\displaystyle + \sum_{k \in G_v^2} c(\epsilon \to k)\ v_k + \sum_{kl \in G_{kl}^2} c(\epsilon \to kl)\ f_{kl}
\end{array} \right) \\
&= \min_{x,y} \left(\begin{array}{l}
\displaystyle\sum_{i \in G_v^1} \sum_{k \in G_v^2} c(i \to k)\ x_{i,k} + \sum_{ij \in G_e^1} \sum_{kl \in G_e^2} c(ij \to kl)\ y_{ij,kl} \\
\displaystyle + \sum_{i \in G_v^1} c(i \to \epsilon)\ (1 - \sum_{k \in G_v^2} x_{i,k}) + \sum_{ij \in G_e^1} c(ij \to \epsilon)\ (1 - \sum_{kl \in G_e^2} y_{ij,kl}) \\
\displaystyle + \sum_{k \in G_v^2} c(\epsilon \to k)\ (1 - \sum_{i \in G_v^1} x_{i,k}) + \sum_{kl \in G_{kl}^2} c(\epsilon \to kl)\ (1 - \sum_{ij \in G_e^1} y_{ij,kl})
\end{array} \right) \\
&= \min_{x,y} \left(\begin{array}{l}
\displaystyle\sum_{i \in G_v^1} \sum_{k \in G_v^2} (c(i \to k) - c(i \to \epsilon) - c(\epsilon \to k))\ x_{i,k} \\
\displaystyle + \sum_{ij \in G_e^1} \sum_{kl \in G_e^2} (c(ij \to kl) - c(ij \to \epsilon) - c(\epsilon \to kl))\ y_{ij,kl} \\
\displaystyle + \sum_{i \in G_v^1} c(i \to \epsilon) + \sum_{k \in G_v^2} c(\epsilon \to k) + \sum_{ij \in G_e^1} c(ij \to \epsilon) + \sum_{kl \in G_{kl}^2} c(\epsilon \to kl)
\end{array} \right)
\end{aligned}
\tag{B.88}
$$

Therefore, based on the defined objective function (*cf.* equation B.88), domain constraints (*cf.* equations (B.89) and (B.90)) and linear inequality constraints (*cf.* equations (B.82), (B.83), (B.84), (B.85), (B.86) and (B.87) in Table B.5), an admissible edit path solution of the optimized BLP that minimizes the objective function applied on the graph G^1 to make it isomorphic to the graph G^2, is given based on the optimized BLP formulation of GED which is illustrated in Table B.6. The optimized BLP formulation of GED has:

- $|G_v^1||G_v^2|+|G_e^1||G_e^2|$ variables,
- $|G_v^1|+|G_v^2|+|G_e^1|+|G_e^2|+|G_v^1||G_e^2|+|G_v^2||G_e^1|$ constraints without taking into consideration the domain constraints (*cf.* equations (B.89) and (B.90)).

where $|G_v^1|$ and $|G_v^2|$ denote the number of vertices of the two graphs G^1 and G^2, respectively. $|G_e^1|$ and $|G_e^2|$ denote the number of edges of the two graphs G^1 and G^2, respectively.

Table B.6.: Optimized BLP formulation of the GED paradigm.

BLP id	**Value**
Objective function	$\min_{x,y} \left(\begin{array}{l} \sum_{i \in G_v^1} \sum_{k \in G_v^2} (c(i \to k) - c(i \to \epsilon) - c(\epsilon \to k))\ x_{i,k} \\ + \sum_{ij \in G_e^1} \sum_{kl \in G_e^2} (c(ij \to kl) - c(ij \to \epsilon) - c(\epsilon \to kl))\ y_{ij,kl} \\ + \sum_{i \in G_v^1} c(i \to \epsilon) + \sum_{k \in G_v^2} c(\epsilon \to k) + \sum_{ij \in G_e^1} c(ij \to \epsilon) + \sum_{kl \in G_{kl}^2} c(\epsilon \to kl) \end{array} \right)$
Linear inequality constraints	$\forall i \in G_v^1,\ \sum_{k \in G_v^2} x_{i,k} \leq 1$
	$\forall k \in G_v^2,\ \sum_{i \in G_v^1} x_{i,k} \leq 1$
	$\forall ij \in G_e^1,\ \sum_{kl \in G_e^2} y_{ij,kl} \leq 1$
	$\forall kl \in G_e^2,\ \sum_{ij \in G_e^1} y_{ij,kl} \leq 1$
	$\forall k \in G_v^2 \forall ij \in G_e^1,\ \sum_{kl \in G_e^2} y_{ij,kl} \leq x_{i,k}$
	$\forall l \in G_v^2 \forall ij \in G_e^1,\ \sum_{kl \in G_e^2} y_{ij,kl} \leq x_{j,l}$
Domain constraints	$\forall (i,k) \in G_v^1 \times G_v^2,\ x_{i,k} \in \{0,1\}$ (B.89)
	$\forall (ij,kl) \in G_e^1 \times G_e^2,\ y_{ij,kl} \in \{0,1\}$ (B.90)

B.10. Computer-aided tool for characterization and categorization of historical book pages

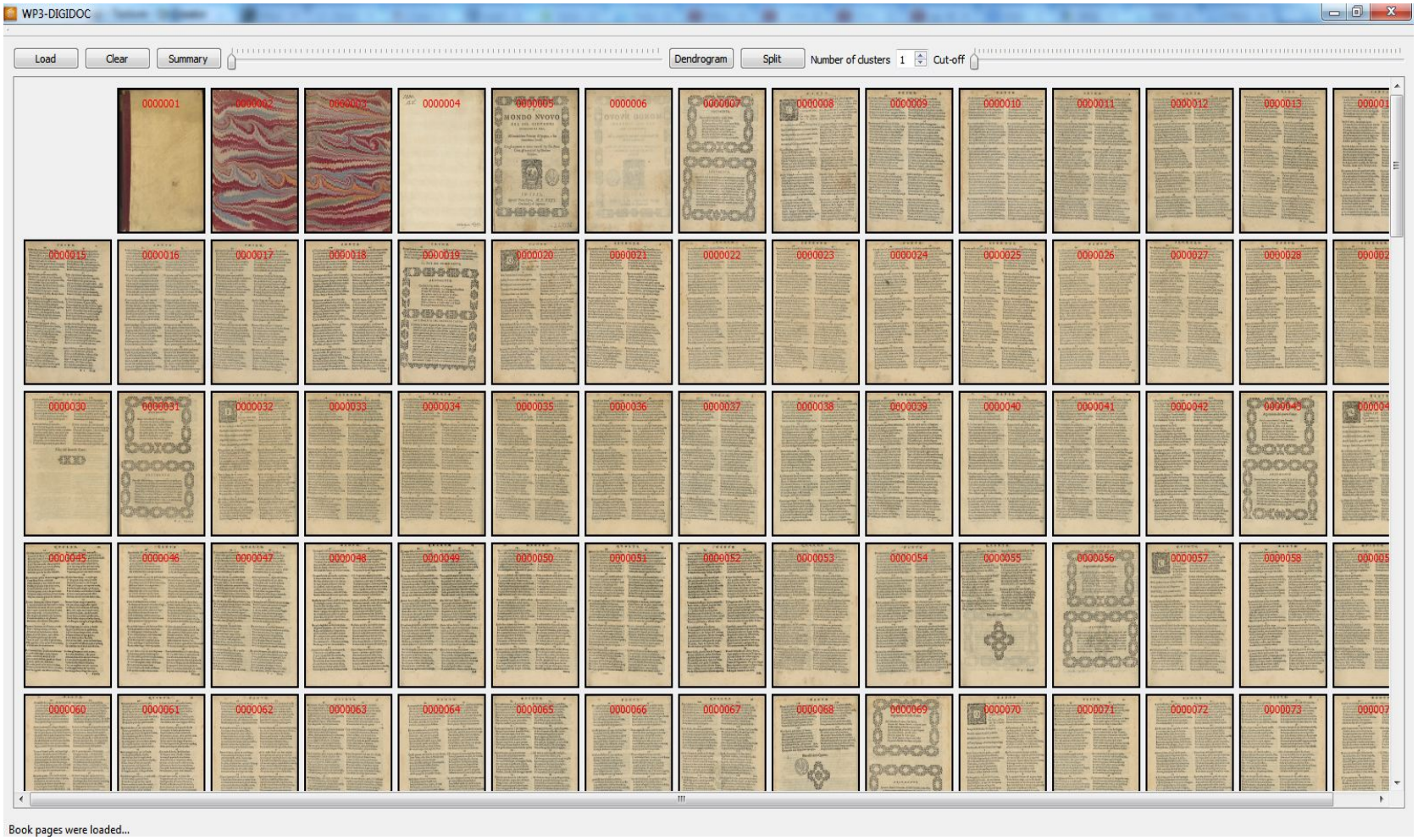


Figure B.38.: GUI Screen shot illustrating the uploading of pages from a DHB directory.

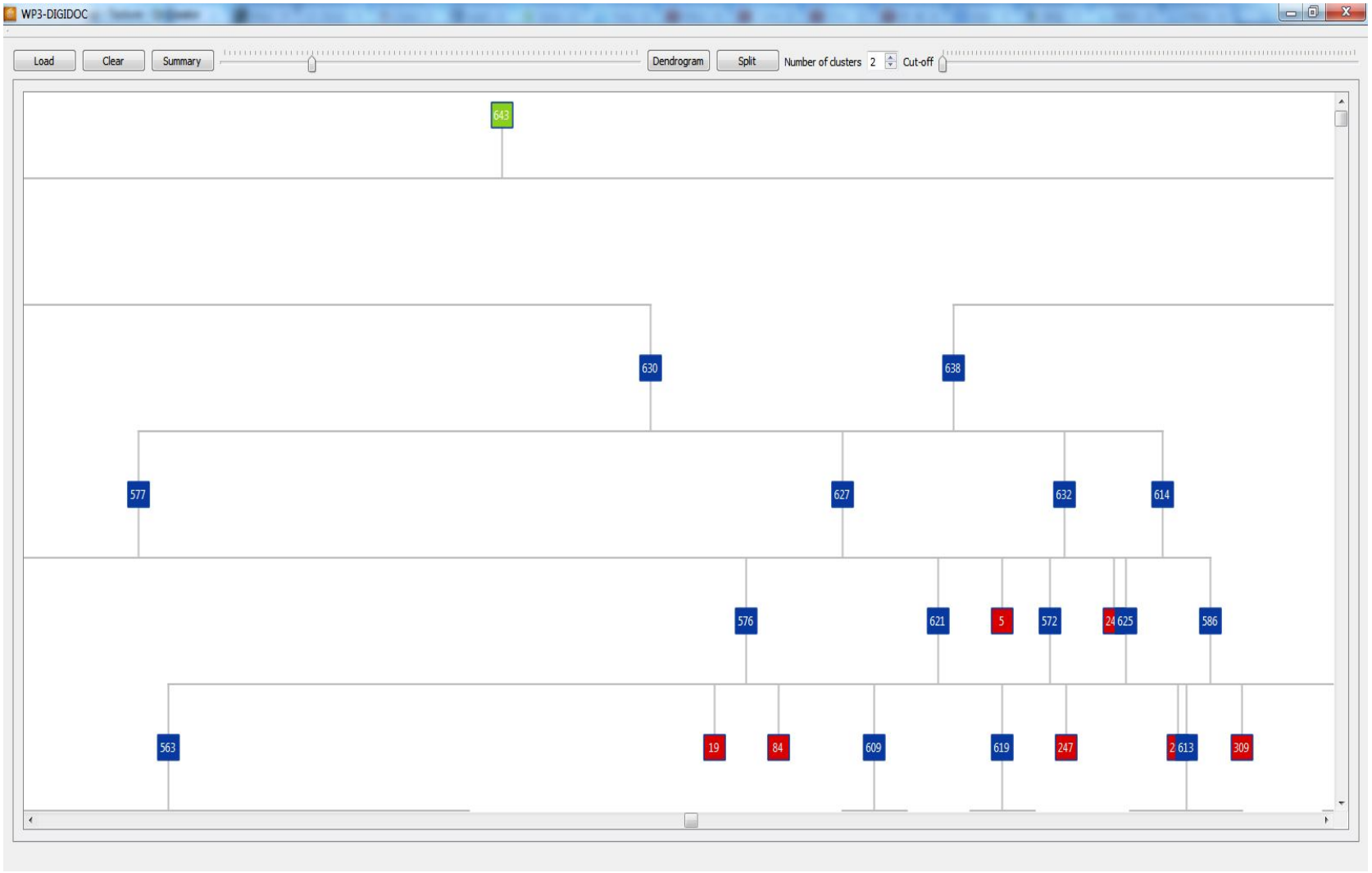


Figure B.39.: GUI Screen shot illustrating the deduced dendrogram from applying an unsupervised classification task (HAC algorithm) which is performed on the obtained distance matrix by computing the dissimilarity between the compared graph-based signatures.

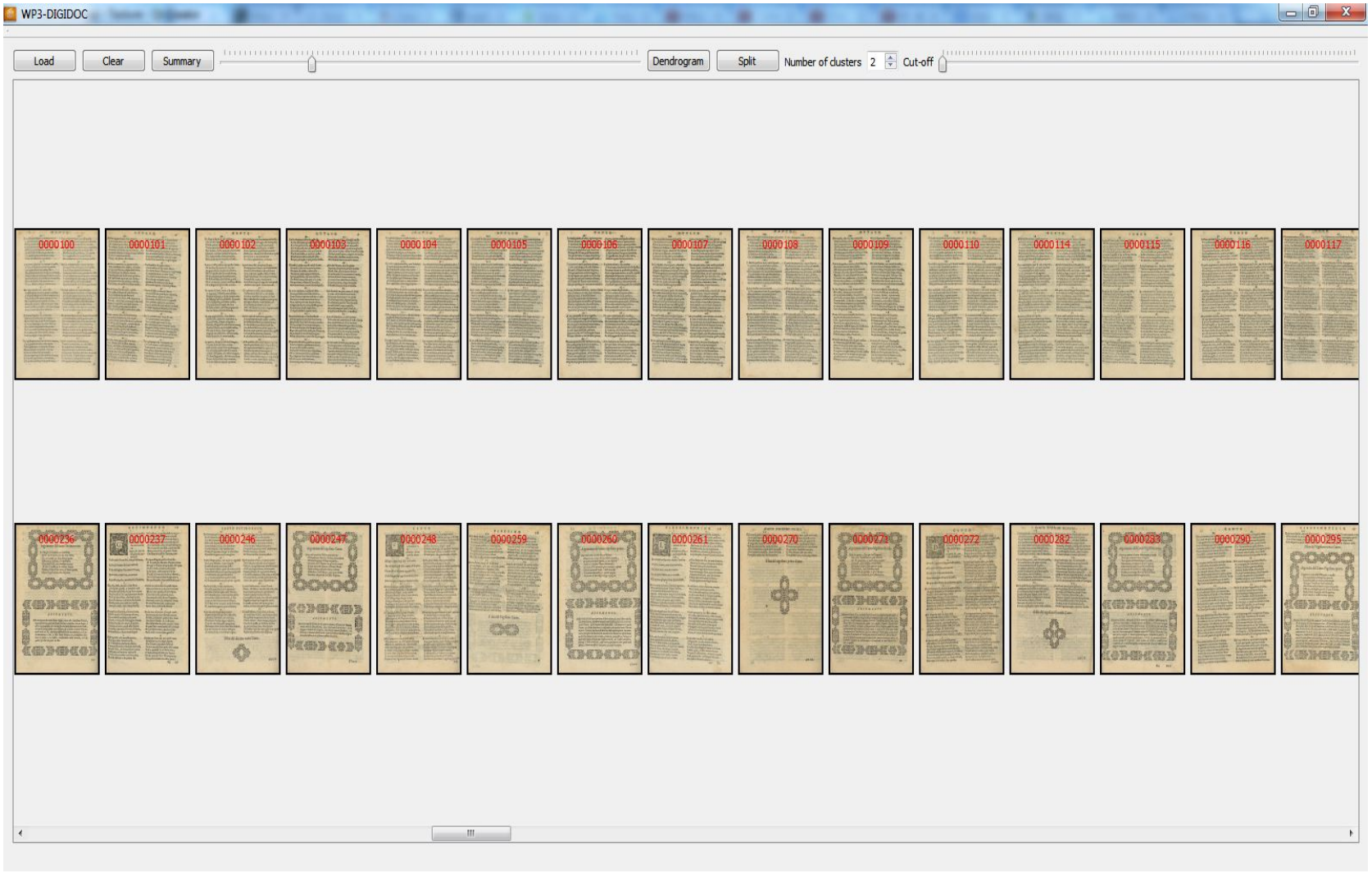


Figure B.40.: GUI Screen shot illustrating the unsupervised classification of the uploaded DHB pages using the HAC algorithm by setting the maximum number of book page types to 2. It shows the separation of the DHB pages into 2 clusters. One cluster representing pages containing only textual regions, and the other one illustrating pages containing textual and graphical regions. Each cluster is represented in a separate line.

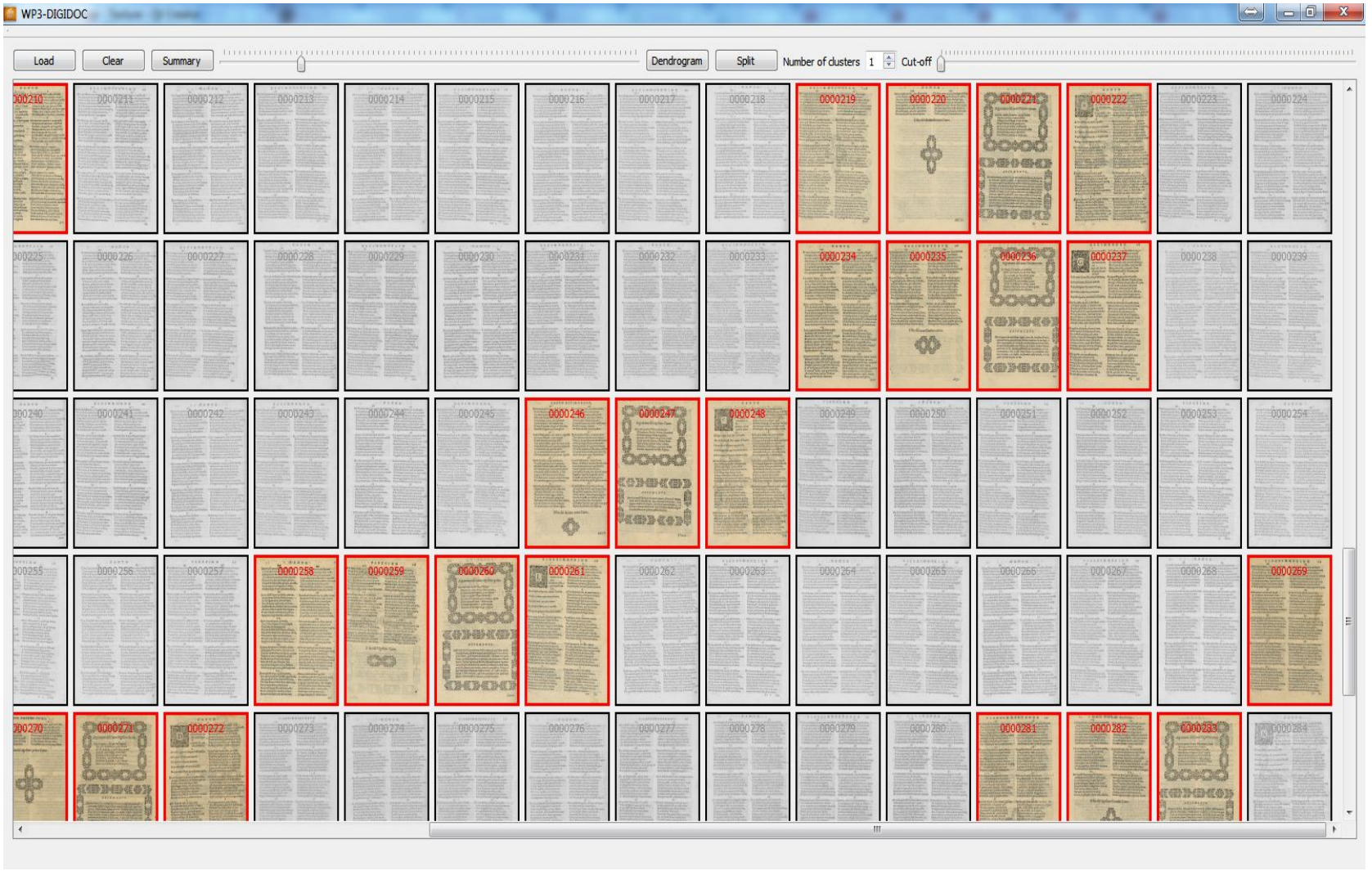


Figure B.41.: GUI Screen shot illustrating an obtained summary of the analyzed DHB. It shows the different detected transition DHB pages. Only DHB pages having GEDs above a pre-defined threshold GED value are retrieved. The shaded DHB pages are considered as non-transition pages, while the DHB pages with red borders are considered as the transition pages (*i.e.* they have layout and/or content that differ from the following page).

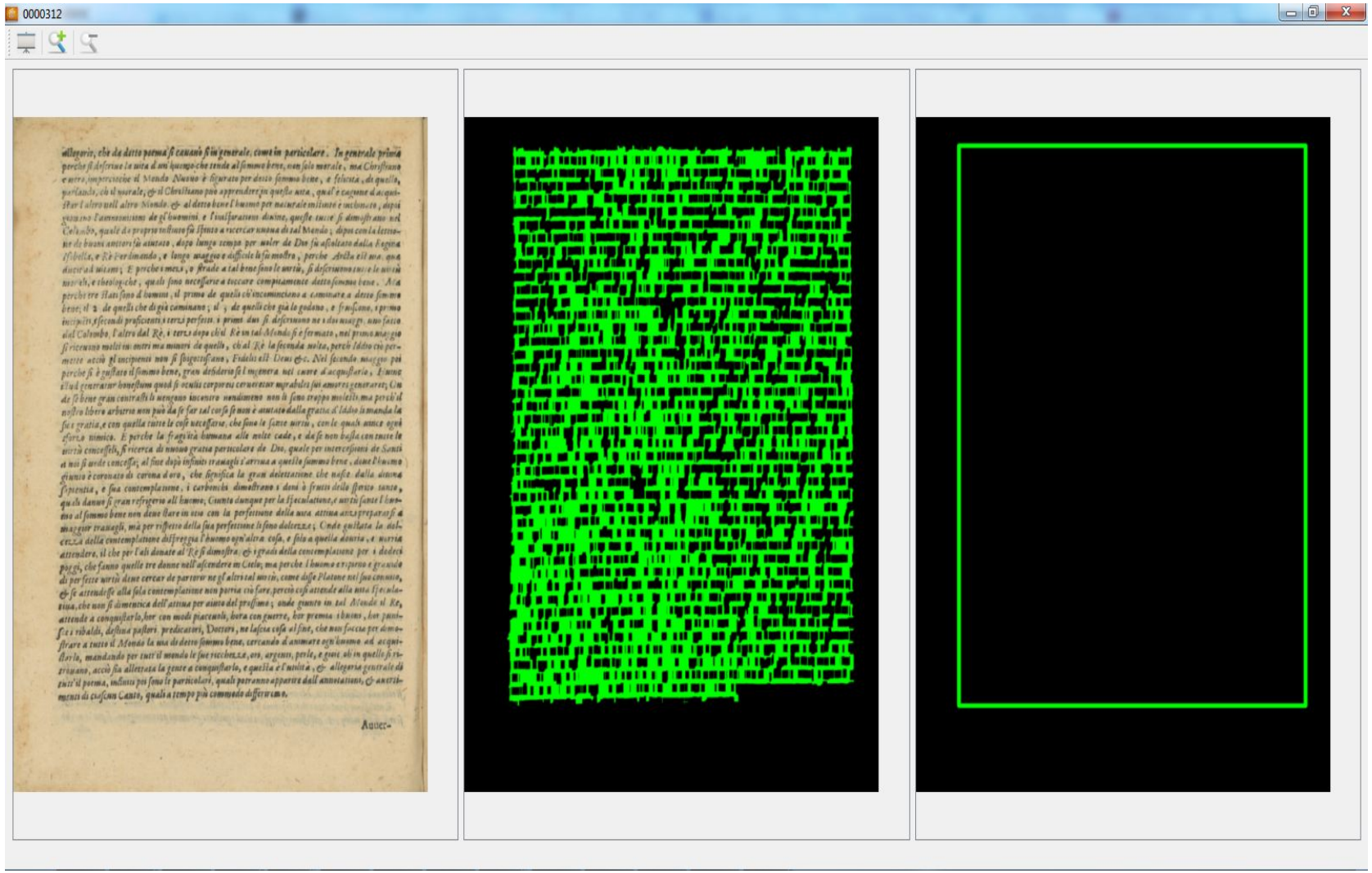


Figure B.42.: GUI Screen shot illustrating an example of the obtained structural signature of a DHB page (containing only text).

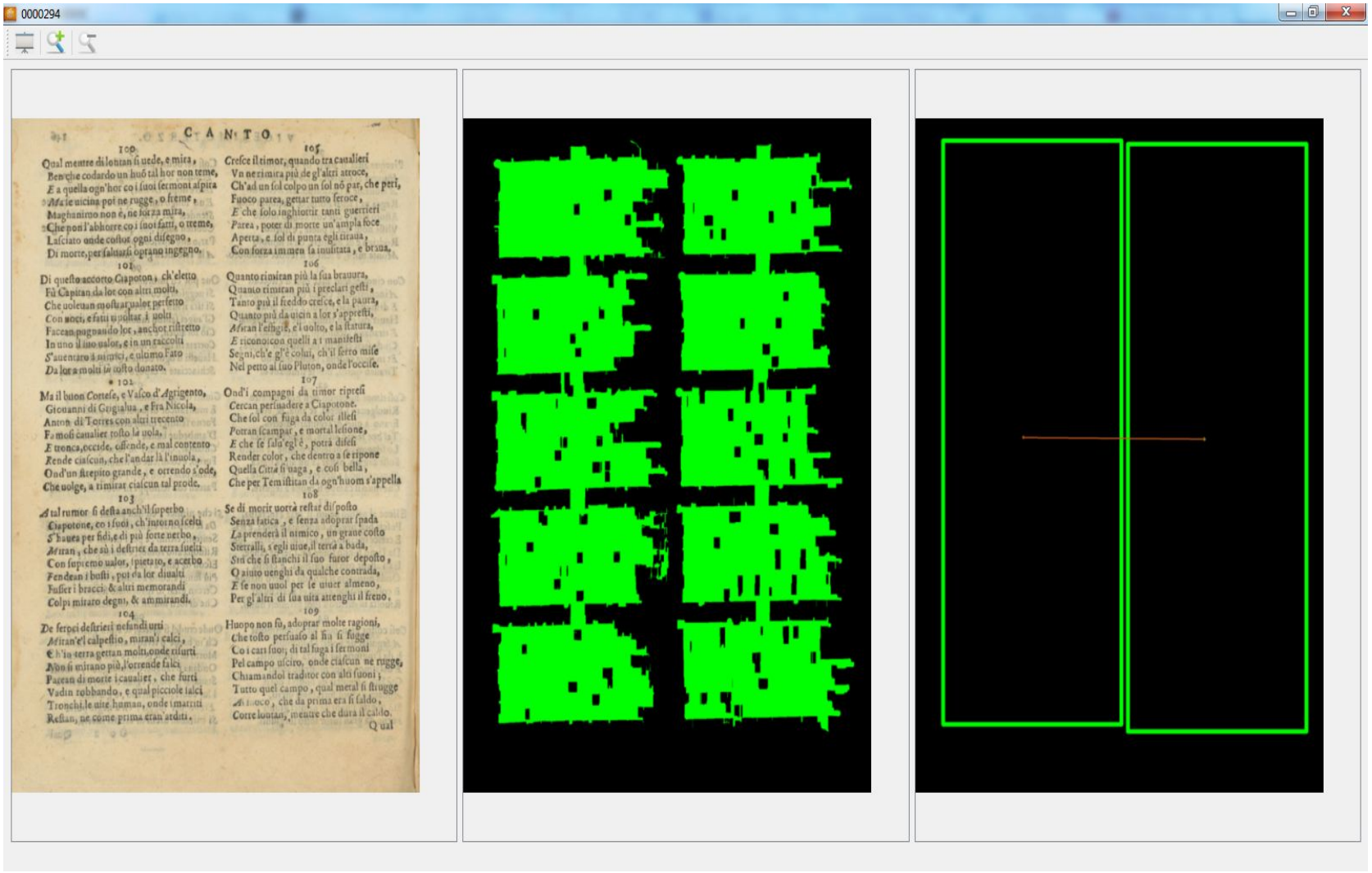


Figure B.43.: GUI Screen shot illustrating an example of the obtained structural signature of a DHB page (containing only text which is presented in two columns).

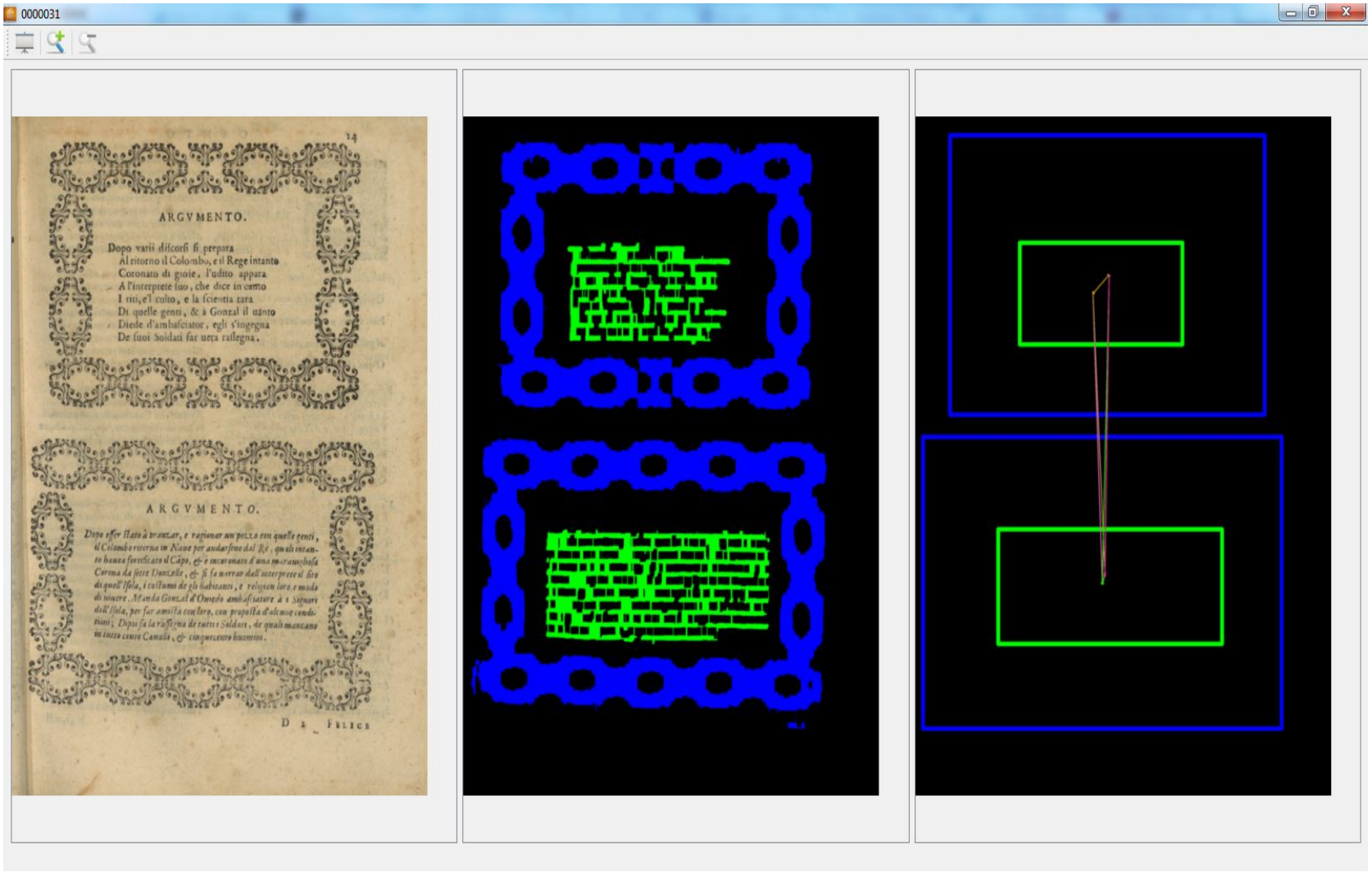


Figure B.44.: GUI Screen shot illustrating an example of the obtained structural signature of a DHB page (containing graphics and text).

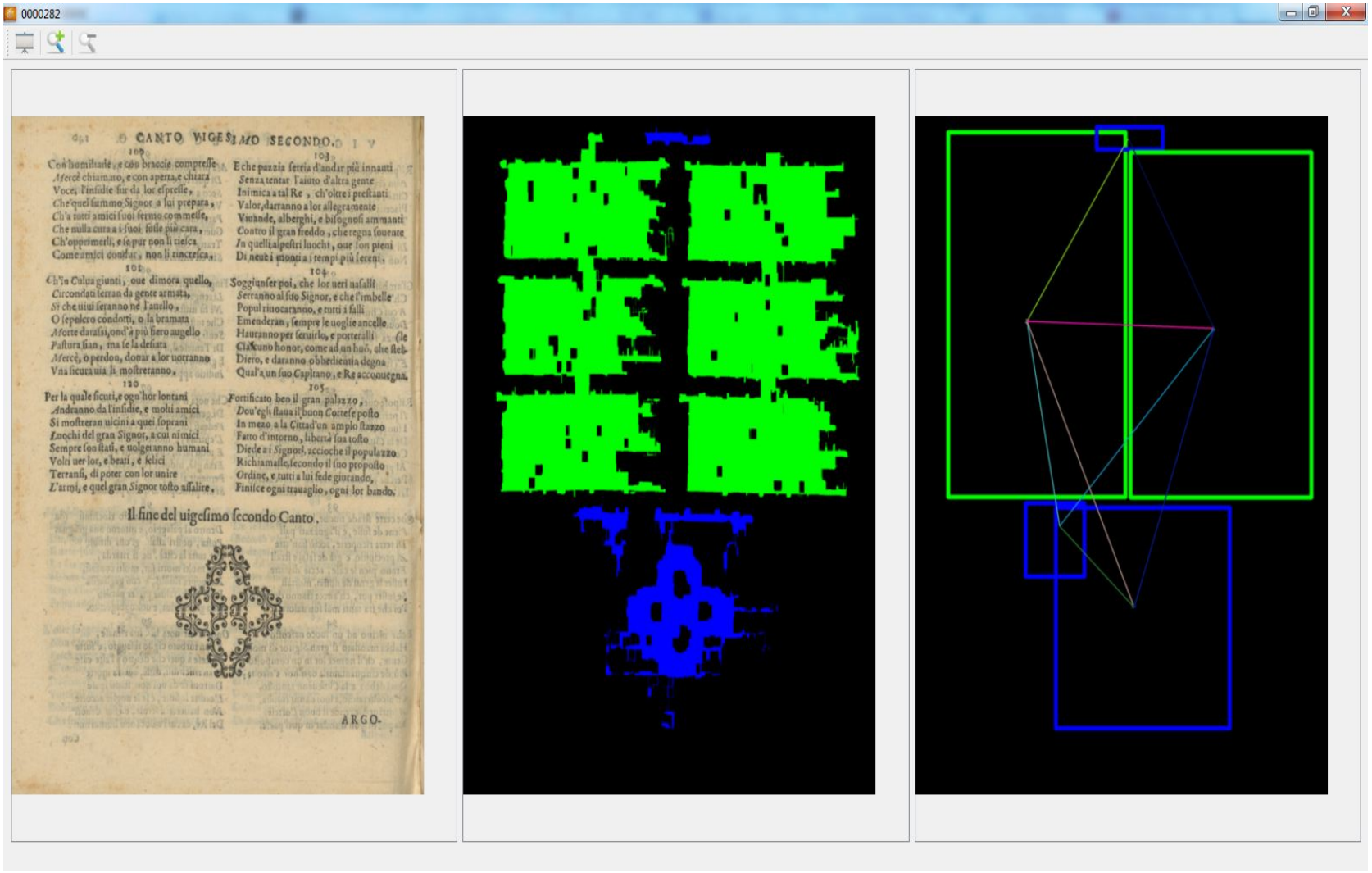


Figure B.45.: GUI Screen shot illustrating an example of the obtained structural signature of a DHB page (containing graphics and text which is presented in two columns).

Bibliography

[1] N. Journet, J. Ramel, R. Mullot, and V. Eglin, "Document image characterization using a multiresolution analysis of the texture: application to old documents," *International Journal of Document Analysis and Recognition*, pp. 9–18, 2008.

[2] F. Nourbakhsh, P. B. Pati, and A. G. Ramakrishnan, "Text localization and extraction from complex gray images," in *Indian Conference on Computer Vision, Graphics and Image Processing*. Springer-Verlag, 2006, pp. 776–785.

[3] M. Cote and A. B. Albu, "Texture sparseness for pixel classification of business document images," *International Journal of Document Analysis and Recognition*, pp. 1–17, 2014.

[4] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 488–493.

[5] K. Kise, *Page segmentation techniques in document analysis*. Handbook of Document Image Processing and Recognition, Springer-Verlag, 2014.

[6] O. Okun and M. Pietikäinen, "A survey of texture-based methods for document layout analysis," in *Workshop on Texture Analysis in Machine Vision*. Springer-Verlag, 1999, pp. 137–148.

[7] H. S. Baird, "Digital libraries and document image analysis," in *International Conference on Document Analysis and Recognition*. IEEE, 2003, pp. 2–14.

[8] M. Coustaty, R. Raveaux, and J. M. Ogier, "Historical document analysis: a review of French projects and open issues," in *European Signal Processing Conference*. EURASIP, 2011, pp. 1445–1449.

[9] F. LeBourgeois, E. Trinh, B. Allier, V. Eglin, and H. Emptoz, "Document images analysis solutions for digital libraries," in *International Workshop on Document Image Analysis for Libraries*. IEEE, 2004, pp. 2–24.

[10] B. Julesz, "Visual pattern discrimination," *Information Theory*, pp. 84–92, 1962.

[11] A. Piper, "Reading's refrain: from bibliography to topology," *Readings: Selected Essays from the English Institute*, pp. 373–399, 2013.

[12] E. T. Nalisnick and H. S. Baird, "Extracting sentiment networks from Shakespeare's plays," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 758–762.

[13] H. Bunke and K. Riesen, "Towards the unification of structural and statistical pattern recognition," *Pattern Recognition Letters*, pp. 811–825, 2012.

[14] H. Bunke, S. Günter, and X. Jiang, "Towards bridging the gap between statistical and structural pattern recognition: two new concepts in graph matching," in *Advances in Pattern Recognition - ICAPR 2001, Lecture Notes in Computer Science*. Springer-Verlag, 2001, pp. 1–11.

[15] S. Jouili, M. Coustaty, S. Tabbone, and J. M. Ogier, "NaviDoMass: structural-based approaches towards handling historical documents," in *International Conference on Pattern Recognition*. IEEE, 2010, pp. 946–949.

[16] J. André and M. A. Chabin, "Les documents anciens," *Document Numérique*, 1999.

[17] J. M. Salaün, "Bibliothèques numériques et google-print," *Regard sur l'actualité*, 2005.

[18] N. Journet, "Analyse d'images de documents anciens : une approche texture," Ph.D. dissertation, University of La Rochelle, La Rochelle, France, 2006.

[19] L. L. Stein and P. J. Lehu, *Literary research and the American realism and naturalism period: strategies and sources*. Scarecrow Press, 2009.

[20] J. M. Ogier, "Ancient document analysis: a set of new research problems," in *Colloque International Francophone sur l'Ecrit et le Document*, 2005.

[21] G. Cron, A. B. Salah, N. Ragot, K. A. Mohand, and T. Paquet, "État de l'art sur la caractérisation d'un document à ocriser," Projet ANR DigiDoc, Tech. Rep. WP7 : Qualité OCR, 2012.

[22] G. Nagy, T. A. Nartker, and S. V. Rice, "Optical character recognition: an illustrated guide to the frontier," in *Document Recognition and Retrieval*. SPIE, 2000.

[23] I. Marosi, "Industrial OCR approaches: architecture, algorithms, and adaptation techniques," in *Document Recognition and Retrieval*. SPIE, 2007.

[24] A. B. Salah, N. Ragot, and T. Paquet, "Adaptive detection of missed text areas in OCR outputs: application to the automatic assessment of OCR quality in mass digitization projects," in *Document Recognition and Retrieval*. SPIE, 2013.

[25] S. V. Rice, J. Kanai, and T. A. Nartker, "The third annual test of OCR accuracy," ISRI, Tech. Rep., 1994.

[26] ——, "The fourth annual test of OCR accuracy," ISRI, Tech. Rep., 1994.

[27] ——, "The fifth annual test of OCR accuracy," ISRI, Tech. Rep., 1994.

[28] S. Vayness and C. Lerouge, "Charte de traitement : OCR brut et HQ, ALTO," Bibliothèque nationale de France, Tech. Rep., 2008.

[29] S. Uttama, P. Loonis, M. Delalandre, and J. M. Ogier, "Segmentation and retrieval of ancient graphic documents," in *International Workshop on Graphics Recognition*. Springer-Verlag, 2006, pp. 88–98.

[30] V. Eglin, S. Bres, and C. Rivero, "Hermite and Gabor transforms for noise reduction and handwriting classification in ancient manuscripts," *International Journal of Document Analysis and Recognition*, pp. 101–122, 2007.

[31] E. Lecolinet, L. Likforman-Sulem, L. Robert, F. Role, and J. L. Lebrave, "An integrated reading and editing environment for scholarly research on literary works and their handwritten sources," in *Conference on Digital Libraries*. ACM, 1998, pp. 144–151.

[32] F. LeBourgeois and H. Emptoz, "DEBORA: digital access to books of the Renaissance," *International Journal of Document Analysis and Recognition*, pp. 193–221, 2007.

[33] A. Antonacopoulos and D. Karatzas, "Document image analysis for world war II personal records," in *International Workshop on Document Image Analysis for Libraries*. IEEE, 2004, pp. 336–341.

[34] M. Baechler, A. Fischer, N. Naji, R. Ingold, H. Bunke, and J. Savoy, "HisDoc: historical document analysis, recognition, and retrieval," in *Digital Humanities - International Conference of the Alliance of Digital Humanities Organizations (ADHO)*, 2012.

[35] S. Calabretto and A. Bozzi, "The philological workstation BAMBI (Better Access to Manuscripts and Browsing of Images)," *J. Digit. Inf.*, 1998.

[36] S. Calabretto, J. M. Pinon, and A. Bozzi, "BAMBI: management system for old manuscripts," *Document Numérique*, pp. 31–50, 1998.

[37] A. Bozzi, "For a digital philology system," *Document Numérique*, pp. 93–101, 1999.

[38] J. M. Ogier and K. Tombre, "Madonne: document image analysis techniques for cultural heritage documents," in *International Conference on Digital Cultural Heritage*, 2006.

[39] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "Historical document layout analysis competition," in *International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1516–1520.

[40] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal of Document Analysis and Recognition*, pp. 139–152, 2007.

[41] A. Fischer, M. Wüthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehhauser, and M. Stolz, "Automatic transcription of handwritten medieval documents," in *International Conference on Virtual Systems and Multimedia*. IEEE, 2009, pp. 137–142.

[42] A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz, "Ground truth creation for handwriting recognition in historical documents," in *International Workshop on Document Analysis Systems*. ACM, 2010, pp. 3–10.

[43] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character HMMs," *Pattern Recognition Letters*, pp. 934–942, 2012.

[44] N. Serrano, F. Castro, and A. Juan, "The RODRIGO database," in *International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), 2010, pp. 2709–2712.

[45] V. Romero, A. Fornés, N. Serrano, J. A. Sánchez, A. H. Toselli, V. Frinken, E. Vidal, and J. Lladós, "The ESPOSALLES database: an ancient marriage license corpus for off-line handwriting recognition," *Pattern Recognition*, pp. 1658–1669, 2013.

[46] D. Fernández-Mota, J. Almazán, N. Cirera, A. Fornés, and J. Lladós, "BH2M: the Barcelona historical handwritten marriages database," in *International Conference on Pattern Recognition*. IEEE, 2014, pp. 256–261.

[47] S. Nicolas, T. Paquet, and L. Heutte, "Enriching historical manuscripts: the Bovary project," in *International Workshop on Document Analysis Systems*. Springer-Verlag, 2004, pp. 135–146.

[48] S. Nicolas, Y. Kessentini, T. Paquet, and L. Heutte, "Handwritten document segmentation using hidden Markov random fields," in *International Conference on Document Analysis and Recognition*. IEEE, 2005, pp. 212–216.

[49] J. Y. Ramel, S. Busson, and M. L. Demonet, "AGORA: the interactive document image analysis tool of the BVH project," in *International Workshop on Document Image Analysis for Libraries*. IEEE, 2006, pp. 145–155.

[50] S. Corsini, "Vers un corpus des ornements typographiques Lausannois du 18 ème siècle : problèmes de définition et de méthode," in *Ornementation typographique et bibliographie historique, actes du Colloque de Mons*, 1988, pp. 139–158.

[51] M. Coustaty, R. Pareti, N. Vincent, and J. M. Ogier, "Towards historical document indexing: extraction of drop cap letters," *International Journal of Document Analysis and Recognition*, pp. 243–254, 2011.

[52] T. M. Rath, R. Manmatha, and V. Lavrenko, "A search engine for historical manuscript images," in *International Conference on Research and Development in Information Retrieval*. ACM, 2004, pp. 369–376.

[53] M. Baechler and R. Ingold, "Multi-resolution layout analysis of Medieval manuscripts using dynamic MLP," in *International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 1185–1189.

[54] A. Fischer, M. Baechler, A. Garz, M. Liwicki, and R. Ingold, "A combined system for text line extraction and handwriting recognition in historical documents," in *International Workshop on Document Analysis Systems*. IEEE, 2014, pp. 71–75.

[55] M. Baechler, M. Liwicki, and R. Ingold, "Text line extraction using DMLP classifiers for historical manuscripts," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1029–1033.

[56] H. Wei, M. Baechler, F. Slimane, and R. Ingold, "Evaluation of SVM, MLP and GMM classifiers for layout analysis of historical documents," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1252–1256.

[57] K. Chen, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold, "Robust text line segmentation for historical manuscript images using color and texture," in *International Conference on Pattern Recognition*. IEEE, 2014, pp. 2978–2983.

[58] H. Wei, K. Chen, R. Ingold, and M. Liwicki, "Hybrid feature selection for historical document layout analysis," in *International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 87–92.

[59] B. Coüasnon, "DMOS, a generic document recognition method: application to table structure analysis in a general and in a specific way," *International Journal of Document Analysis and Recognition*, pp. 111–122, 2006.

[60] B. Coüasnon, J. Camillerapp, and I. Leplumey, "Access by content to handwritten archive documents: generic document recognition method and platform for annotations," *International Journal of Document Analysis and Recognition*, pp. 223–242, 2007.

[61] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," *Pattern Recognition*, pp. 2080–2092, 2005.

[62] V. Eglin, D. Gaceb, H. Daher, S. Bres, and N. Vincent, "Outils d'analyse de la dynamique des écritures médiévales. pour l'aide à l'expertise paléographique," *Document Numérique*, pp. 81–104, 2011.

[63] H. Daher, D. Gaceb, V. Eglin, S. Bres, and N. Vincent, "Unsupervised categorization method of graphemes on handwritten manuscripts: application to style recognition," in *Document Recognition and Retrieval*. SPIE, 2012.

[64] F. Cruz-Fernández and O. Ramos-Terrades, "Document segmentation using relative location features," in *International Conference on Pattern Recognition*. IEEE, 2012, pp. 1562–1565.

[65] F. Álvaro, F. Cruz, J. A. Sánchez, O. Ramos-Terrades, and J. M. Benedí, "Page segmentation of structured documents using 2-D stochastic context-free grammars," in *Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*. Springer-Verlag, 2013.

[66] D. Fernández-Mota, J. Lladós, and A. Fornés, "A graph-based approach for segmenting touching lines in historical handwritten documents," *International Journal of Document Analysis and Recognition*, pp. 293–312, 2014.

[67] D. Fernández-Mota, J. Lladós, A. Fornés, and R. Manmatha, "Sequential word spotting in historical handwritten documents," in *International Workshop on Document Analysis Systems*. IEEE, 2014, pp. 101–105.

[68] J. Bigün, S. K. Bhattacharjee, and S. Michel, "Orientation radiograms for image retrieval: an alternative to segmentation," in *International Conference on Pattern Recognition*. IEEE, 1996, pp. 346–350.

[69] D. Droixhe, S. Stiennon, and N. Vanwelkenhuyzen, "Le projet môriane," in *Vers une nouvelle érudition : numérisation et recherche en histoire du livre, Rencontres Jacques Cartier*, 1999, pp. 139–158.

[70] S. Corsini, "Les ornements des imprimeurs de l'ancien temps sur le web," in *Vers une nouvelle érudition : numérisation et recherche en histoire du livre, Rencontres Jacques Cartier*, 1999, pp. 139–158.

[71] G. Nagy, "Twenty years of document image analysis in PAMI," *Pattern Analysis and Machine Intelligence*, pp. 38–62, 2000.

[72] J. Y. Ramel, S. Leriche, M. L. Demonet, and S. Busson, "User-driven page layout analysis of historical printed books," *International Journal of Document Analysis and Recognition*, pp. 243–261, 2007.

[73] B. Coüasnon, J. Camillerapp, and I. Leplumey, "Making handwritten archives documents accessible to public with a generic system of document image analysis," in *International Workshop on Document Image Analysis for Libraries*. IEEE, 2004, pp. 270–277.

[74] C. Gravenhorst, "Making the past a thing of the future: automated workflow for the conversion of printed items into fully structured digital objects based on common open metadata standards," in *Computers Helping People with Special Needs, Lecture Notes in Computer Science*, 2006, pp. 92–95.

[75] D. Hebert, T. Palfray, S. Nicolas, P. Tranouez, and T. Paquet, "Automatic article extraction in old newspapers digitized collections," in *International Conference on Digital Access to Textual Cultural Heritage*. ACM, 2014, pp. 3–8.

[76] ——, "PIVAJ: displaying and augmenting digitized newspapers on the web experimental feedback from the "Journal de Rouen" collection," in *International Conference on Digital Access to Textual Cultural Heritage*. ACM, 2014, pp. 173–178.

[77] N. Naji and J. Savoy, "Information retrieval strategies for digitized handwritten medieval documents," in *Asia Information Retrieval Societies Conference*. Information Retrieval Technology, Lecture Notes in Computer Science, 2011, pp. 103–114.

[78] J. Savoy and N. Naji, "Comparative information retrieval evaluation for scanned documents," in *WSEAS International Conference on Computers*. World Scientific and Engineering Academy and Society, 2011, pp. 527–534.

[79] N. Journet, J. Ramel, V. Eglin, and R. Mullot, "Caractérisation de la mise en page des documents imprimés de la renaissance par une analyse des orientations," in *Colloque du Groupe d'Études du Traitement du Signal et des Images (GRETSI)*, 2005, pp. 122–129.

[80] M. Cheriet, R. F. Moghaddam, and R. Hedjam, "Visual language processing (VLP) of ancient manuscripts: converting collections to windows on the past," in *GCC Conference and Exhibition*. IEEE, 2013, pp. 407–412.

[81] R. J. Qureshi, J. Y. Ramel, D. Barret, and H. Cardot, "Symbol spotting in graphical documents using graph representations," in *International Workshop on Graphics Recognition*. Springer-Verlag, 2007, pp. 91–103.

[82] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Lladós, and A. Fornés, "A coarse-to-fine word spotting approach for historical handwritten documents based on graph embedding and graph edit distance," in *International Conference on Pattern Recognition*. IEEE, 2014, pp. 3074–3079.

[83] S. Jouili, "Indexation de masses de documents graphiques : approches structurelles," Ph.D. dissertation, University of Nancy 2, Nancy, France, 2011.

[84] J. P. Salmon, L. Wendling, and S. Tabbone, "Improving the recognition by integrating the combination of descriptors," *International Journal of Document Analysis and Recognition*, pp. 3–12, 2007.

[85] S. Tabbone and D. Zuwala, "An indexing method for graphical documents," in *International Conference on Document Analysis and Recognition*. IEEE, 2007, pp. 789–793.

[86] D. W. Embley, S. Machado, T. Packer, J. Park, A. Zitzelberger, S. W. Liddle, and D. W. Lonsdale, "Enabling search for facts and implied facts in historical documents," in *International Workshop on Historical Document Imaging and Processing*. ACM, 2011, pp. 59–66.

[87] T. L. Packer and D. W. Embley, "Cost effective ontology population with data from lists in ocred historical documents," in *International Workshop on Historical Document Imaging and Processing*. ACM, 2013, pp. 44–52.

[88] Y. Liang, R. M. Guest, M. C. Fairhurst, L. Heutte, S. Nicolas, A. Burnett, and T. Palfray, "EMMEL: a framework for historical manuscript analysis and presentation," *Universal Access in the Information Society*, pp. 147–160, 2014.

[89] C. Grana, G. Serra, M. Manfredi, D. Coppi, and R. Cucchiara, "Layout analysis and content enrichment of digitized books," *Multimedia Tools and Applications*, pp. 1–22, 2014.

[90] S. Uttama, J. M. Ogier, and P. Loonis, "Top-down segmentation of ancient graphical drop caps: lettrines," in *International Workshop on Graphics Recognition*. Springer-Verlag, 2005, pp. 87–96.

[91] Y. Y. Tang, S. W. Lee, and C. Y. Suen, "Automatic document processing: a survey," *Pattern Recognition*, pp. 1931–1952, 1996.

[92] S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms: a literature survey," in *Document Recognition and Retrieval*. SPIE, 2003, pp. 197–207.

[93] M. Diem and R. Sablatnig, "Recognition of degraded handwritten characters using local features," in *International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 221–225.

[94] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal of Document Analysis and Recognition*, pp. 123–138, 2007.

[95] R. Mullot, *Les documents écrits : de la numérisation à l'indexation par le contenu.* Hermès, 2006.

[96] O. Augereau, N. Journet, A. Vialard, and J. P. Domenger, "Improving classification of an industrial document image database by combining visual and textual features," in *International Workshop on Document Analysis Systems.* IEEE, 2014, pp. 314–318.

[97] M. R. Bouguelia, Y. Belaid, and A. Belaid, "A stream-based semi-supervised active learning approach for document classification," in *International Conference on Document Analysis and Recognition.* IEEE, 2013, pp. 611–615.

[98] B. Klein, A. R. Dengel, and A. Fordan, "smartFIX: an adaptive system for document analysis and understanding," in *Reading and Learning, Lecture Notes in Computer Science.* Springer-Verlag, 2004, pp. 166–186.

[99] G. Agam, G. Bal, G. Frieder, and O. Frieder, "Degraded document image enhancement," in *Document Recognition and Retrieval.* SPIE, 2007.

[100] L. Likforman-Sulem, "Apport du traitement des images à la numérisation des documents anciens," *Document Numérique*, pp. 13–26, 2003.

[101] K. Wong, R. Casey, and F. Wahl, "Document analysis system," *IBM Journal of Research and Development*, pp. 647–656, 1982.

[102] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, pp. 375–390, 1982.

[103] L. O'Gorman, "The document spectrum for page layout analysis," *Pattern Analysis and Machine Intelligence*, pp. 1162–1173, 1993.

[104] G. Nagy and S. Seth, "Hierarchical representation of optically scanned documents," in *International Conference on Pattern Recognition.* IEEE, 1984, pp. 347–349.

[105] T. Pavlidis and J. Zhou, "Page segmentation and classification," *Graphical Model and Image Processing*, pp. 484–496, 1992.

[106] H. Kida, . Iwaki, and K. Kawada, "Document recognition system for office automation," in *International Conference on Pattern Recognition.* IEEE, 1986, pp. 446–448.

[107] K. Chen, F. Yin, and C. L. Liu, "Hybrid page segmentation with efficient whitespace rectangles extraction and grouping," in *International Conference on Document Analysis and Recognition.* IEEE, 2013, pp. 958–962.

[108] G. Lazzara, R. Levillain, T. Geraud, Y. Jacquelet, J. Marquegnies, and A. Crépin-Leblond, "The SCRIBO module of the Olena platform: a free software framework for document image analysis," in *International Conference on Document Analysis and Recognition.* IEEE, 2011, pp. 252–258.

[109] H. S. Baird, "Anatomy of a versatile page reader," *Proceedings of the IEEE*, pp. 1059–1065, 1992.

[110] T. M. Breuel, "Two geometric algorithms for layout analysis," in *Document Analysis Systems.* Springer-Verlag, 2002, pp. 188–199.

[111] A. Antonacopoulos, "Page segmentation using the description of the background," *Computer Vision and Image Understanding*, pp. 350–369, 1998.

[112] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area Voronoi diagram," *Computer Vision and Image Understanding*, pp. 370–382, 1998.

[113] M. Agrawal and D. Doermann, "Context-aware and content-based dynamic Voronoi page segmentation," in *International Workshop on Document Analysis Systems*. ACM, 2010, pp. 73–80.

[114] H. S. Baird, "Background structure in document images," in *H. Bunke, P. S. P. Wang and H. S. Baird (Eds.), Document Image Analysis*. World Scientific, 1994, pp. 17–34.

[115] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals," *Pattern Analysis and Machine Intelligence*, pp. 737–743, 1993.

[116] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *Pattern Analysis and Machine Intelligence*, pp. 910–918, 1988.

[117] S. N. Srihari and V. Govindaraju, "Analysis of textual images using the Hough transform," in *Machine Vision and Applications*. Springer-Verlag, 1989, pp. 141–153.

[118] D. S. Bloomberg, "Textured reductions for document image analysis," in *Document Recognition and Retrieval*. SPIE, 1996, pp. 160–174.

[119] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Improved document image segmentation algorithm using multiresolution morphology," in *Document Recognition III*. SPIE, 2011, pp. 1–100.

[120] D. J. Ittner and H. S. Baird, "Language-free layout analysis," in *International Conference on Document Analysis and Recognition*. IEEE, 1993, pp. 336–340.

[121] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data structures and algorithms*. Reading, Mass.: Addison-Wesley Publishing Company, 1983.

[122] A. P. Dias, "Minimum spanning trees for text segmentation," in *Symposium on Document Analysis and Recognition*. SPIE, 1995, pp. 51–65.

[123] A. Simon, J. C. Pret, and A. P. Johnson, "A fast algorithm for bottom-up document layout analysis," *Pattern Analysis and Machine Intelligence*, pp. 273–277, 1997.

[124] K. Kise, M. Iwata, A. Dengel, and K. Matsumoto, "Text-line extraction as selection of paths in the neighbor graph," in *International Workshop on Document Analysis Systems*. Springer-Verlag, 1998, pp. 225–239.

[125] Y. P. Zhou and C. L. Tan, "Hough technique for bar charts detection and recognition in document images," in *International Conference on Image Processing*. IEEE, 2000, pp. 605–608.

[126] J. He and A. C. Downton, "User-assisted archive document image analysis for digital library construction," in *International Conference on Document Analysis and Recognition*. IEEE, 2003, pp. 498–502.

[127] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos, "Segmentation of historical machine-printed documents using adaptive run-length smoothing and skeleton segmentation paths," *Image and Vision Computing*, pp. 590–604, 2010.

[128] A. Belaïd and N. Ouwayed, *Guide to OCR for Arabic scripts: segmentation of ancient Arabic documents.* Springer, 2011.

[129] V. Malleron, V. Eglin, H. Emptoz, S. Dord-Crouslé, and P. Régnier, "Text lines and snippets extraction for 19th century handwriting documents layout analysis," in *International Conference on Document Analysis and Recognition.* IEEE, 2009, pp. 1001–1005.

[130] J. Serra, *Image analysis and mathematical morphology.* Academic Press, 1982.

[131] I. Granado, M. Mengucci, and F. Muge, "Extraction de textes et de figures dans les livres anciens à l'aide de la morphologie mathématique," in *Colloque International Francophone sur l'Ecrit et le Document*, 2000.

[132] F. Muge, I. Granado, M. Mengucci, P. Pina, V. Ramos, N. Sirakov, J. R. C. Pinto, A. Marcolino, M. Ramalho, P. Vieira, and A. M. d. Amaral, "Automatic feature extraction and recognition for digital access of books of the Renaissance," in *Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science.* Springer-Verlag, 2000, pp. 1–13.

[133] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of historical handwritten documents into text zones and text lines," in *International Conference on Frontiers in Handwriting Recognition.* IEEE, 2014, pp. 464–469.

[134] Z. Shi and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," in *International Workshop on Document Image Analysis for Libraries.* IEEE, 2004, pp. 306–312.

[135] J. André, H. Richy, L. Likforman-Sulem, and G. Ventabert, "Electronic representation and use of old documents (texts and images): about Philectre project experiments," *Document Numérique*, pp. 57–73, 1999.

[136] A. Antonacopoulos, B. Gatos, and D. Bridson, "Page segmentation competition," in *International Conference on Document Analysis and Recognition.* IEEE, 2007, pp. 1279–1283.

[137] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos, "ICDAR 2009 page segmentation competition," in *International Conference on Document Analysis and Recognition.* IEEE, 2009, pp. 1370–1374.

[138] F. Shafait, D. Keysers, and T. M. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *Pattern Analysis and Machine Intelligence*, pp. 941–954, 2008.

[139] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *Computer*, pp. 10–22, 1992.

[140] S. Mao and T. Kanungo, "Empirical performance evaluation methodology and its application to page segmentation algorithms," *Pattern Analysis and Machine Intelligence*, pp. 242–256, 2001.

[141] H. Wechsler, "Texture analysis - a survey," *Signal Processing*, pp. 271–282, 1980.

[142] T. R. Reed and J. M. H. DuBuf, "A review of recent texture segmentation and feature extraction techniques," *CVGIP: Image Understanding*, pp. 359–372, 1993.

[143] Y. Liua, S. Wub, and X. Zhoua, "Texture segmentation based on features in wavelet domain for image retrieval," pp. 2026–2034, 2003.

[144] W. K. Pratt, O. D. Faugeras, and A. Gagalowicz, "Visual discrimination of stochastic texture fields," *Systems Man and Cybernetics*, pp. 796–804, 1978.

[145] R. M. Haralick, "Statistical and structural approaches to texture," *In Proceedings of the IEEE*, pp. 786–804, 1979.

[146] R. M. Pickett, *Visual analysis of texture in the detection and recognition of objects.* Picture Processing and Psychopictorics, Academic Press, 1970.

[147] J. K. Hawkins, *Textural properties for pattern recognition.* Picture Processing and Psychopictorics, Academic Press, 1970.

[148] B. Julesz, *Preconscious and conscious processes in vision.* Bell Laboratories. Murray Hill, New Jersey. Pattern Recognition Mechanisms, 1985.

[149] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *Pattern Analysis and Machine Intelligence*, pp. 4–37, 2000.

[150] S. W. Zucker, "Toward a model of texture," *Computer Graphics and Image Processing*, pp. 190–202, 1976.

[151] T. Toyoda and O. Hasegawa, "Texture classification using extended higher order local autocorrelation features," in *International Workshop on Texture Analysis and Synthesis*, 2005, pp. 131–136.

[152] R. Chellappa and S. Chatterjee, "Classification of textures using Markov random field models," in *International Conference on Acoustics, Speech, and Signal Processing.* IEEE, 1984, pp. 694–697.

[153] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence*, pp. 971–987, 2002.

[154] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *Pattern Analysis and Machine Intelligence*, pp. 674–693, 1989.

[155] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *Pattern Analysis and Machine Intelligence*, pp. 837–842, 1996.

[156] N. Feddaoui and K. Hamrouni, "Personal identification based on texture analysis of Arabic handwriting text," in *International Conference on Information & Communication Technologies.* IEEE, 2006, pp. 1302–1307.

[157] J. Zhang and T. Tan, "Brief review of invariant texture analysis methods," *Pattern Recognition*, pp. 735–747, 2002.

[158] K. I. Laws, "Textured image segmentation," University of Southern California, Los Angeles, Image processing Institute, Tech. Rep. USCIPI Report 940, 1980.

[159] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *Systems Man and Cybernetics*, pp. 460–473, 1978.

[160] F. D'Astous and M. E. Jernigan, "Texture discrimination based on detailed measures of the power spectrum," in *International Conference on Pattern Recognition.* IEEE, 1984, pp. 83–86.

[161] R. L. Kashyap and A. Khotanzad, "A model-based method for rotation invariant texture classification," *Pattern Analysis and Machine Intelligence*, pp. 472–481, 1986.

[162] J. M. Francos, A. Narasimhan, and J. W. Woods, "Maximum likelihood parameter estimation of textures using a Wold-decomposition based model," *Image Processing*, pp. 1655–1666, 1995.

[163] H. Greenspan, S. Belongie, R. Goodman, and P. Perona, "Rotation invariant texture recognition using a steerable pyramid," in *International Conference on Pattern Recognition.* IEEE, 1984, pp. 162–167.

[164] R. K. Goyal, W. L. Goh, D. P. Mital, and K. L. Chan, "A translation rotation and scale invariant texture analysis technique based on structural properties," in *International Conference on Automation Technology.* IEEE, 1994.

[165] ——, "Scale and rotation invariant texture analysis based on structural property," in *International Conference on Industrial Electronics, Control, and Instrumentation.* IEEE, 1995, pp. 1290–1294.

[166] G. Eichmann and T. Kasparis, "Topologically invariant texture descriptors," *Computer Vision, Graphics, and Image Processing*, pp. 267–281, 1988.

[167] W. K. Lam and C. K. Li, "Rotated texture classification by improved iterative morphological decomposition," *Vision, Image and Signal Processing*, pp. 171–179, 1997.

[168] J. Li, J. Z. Wang, and G. Wiederhold, "Classification of textured and non-textured images using region segmentation," *Image Processing*, pp. 754–757, 2000.

[169] D. Wang and S. N. Srihari, "Page segmentation and classification," *Computer Vision, Graphics, and Image Processing*, pp. 327–352, 1989.

[170] D. Chetverikov, J. Liang, J. Kömüves, and R. M. Haralick, "Zone classification using texture features," in *International Conference on Pattern Recognition.* IEEE, 1996, pp. 676–680.

[171] V. Eglin and A. Gagneux, "Visual exploration and functional document labeling," in *International Conference on Document Analysis and Recognition.* IEEE, 2001, pp. 816–820.

[172] B. Allier, J. Duong, A. Gagneux, P. Mallet, and H. Emptoz, "Texture feature characterization for logical pre-labeling," in *International Conference on Document Analysis and Recognition.* IEEE, 2003, pp. 567–571.

[173] J. S. Payne, T. J. Stonham, and D. Patel, "Document segmentation using texture analysis," in *International Conference on Pattern Recognition.* IEEE, 1994, pp. 380–382.

[174] J. L. Chen, "A simplified approach to the HMM based texture analysis and its application to document segmentation," *Pattern Recognition Letters*, pp. 993–1007, 1997.

[175] B. R. Kim and W. H. Kim, "Texture-based PCA for classifying contents in document image," in *International Conference on Image Processing, Computer Vision, and Pattern Recognition.* CSREA Press, 2008, pp. 228–233.

[176] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis.* John Wiley & Sons, 1990.

[177] C. H. Chen, L. F. Pau, and P. Wang, *Texture analysis in the handbook of pattern recognition and computer vision*, 2nd ed. World Scientific, 1998.

[178] M. Tuceryan and A. K. Jain, *Texture analysis.* The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), World Scientific Publishing Co, 1998.

[179] M. Petrou and P. G. Sevilla, *Image processing: dealing with texture.* John Wiley & Sons, 2006.

[180] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Systems Man and Cybernetics*, pp. 610–621, 1973.

[181] M. M. Galloway, "Texture analysis using gray level run lengths," *Computer Graphics and Image Processing*, pp. 172–179, 1975.

[182] M. Hall-Beyer. (2000) GLCM texture: a tutorial. National Council on Geographic Information and Analysis Remote Sensing Core Curriculum. [Online]. Available: http://www.fp.ucalgary.ca/mhallbey/tutorial.htm

[183] L. Caponetti, C. Castiello, and P. Górecki, "Document page segmentation using neuro-fuzzy approach," *Applied Soft Computing*, pp. 118–126, 2008.

[184] M. Tuceryan and A. K. Jain, "Texture segmentation using Voronoi polygons," *Pattern Analysis and Machine Intelligence*, pp. 211–216, 1990.

[185] M. Tuceryan, "Moment based texture segmentation," *Pattern Recognition Letters*, pp. 659–668, 1994.

[186] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning.* Morgan Kaufmann Publishers Inc, 2001, pp. 282–289.

[187] R. Ferrell, S. Gleason, and K. Tobin, "Application of fractal encoding techniques for image segmentation," in *International Conference on Quality Control by Artificial Vision.* SPIE, 2003, pp. 69–77.

[188] A. K. Jain, S. K. Bkattacharjee, and Y. Chen, "On texture in document images," in *Computer Vision and Pattern Recognition.* IEEE, 1992, pp. 677–680.

[189] A. K. Jain and S. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing," *Machine Vision and Applications*, pp. 169–184, 1992.

[190] C. Sabharwal and S. Subramanya, "Indexing image databases using wavelet and discrete Fourier transform," in *Symposium on Applied Computing.* ACM, 2001, pp. 434–439.

[191] S. Raju, P. Pati, and A. Ramakrishnan, "Text localization and extraction from complex color images," in *International Symposium on Visual Computing.* Springer-Verlag, 2005, pp. 486–493.

[192] Y. Qiao, Z. Lu, C. Song, and S. Sun, "Document image segmentation using Gabor wavelet and kernel-based methods," in *International Symposium on Systems and Control in Aerospace and Astronautics.* IEEE, 2006, pp. 450–455.

[193] K. Varshney, "Block-segmentation and classification of grayscale postal images," Report in School of Electrical and Computer Engineering, Cornell University, Tech. Rep., 2004.

[194] V. Eglin, S. Bres, and H. Emptoz, "Characterization and classification of printed text in a multiscale context," in *International workshop on structural and syntactic pattern recognition.* Springer-Verlag, 1998, pp. 960–967.

[195] J. Sauvola and M. Pietikäinen, "Skew angle detection using texture direction analysis," in *Scandinavian Conference on Image Analysis.* Springer-Verlag, 1995, pp. 1099–1106.

[196] Y. Liu and S. N. Srihari, "Document image binarization based on texture features," *Pattern Analysis and Machine Intelligence*, pp. 540–544, 1997.

[197] C. L. Liu, M. Koga, and H. Fujisawa, "Gabor feature extraction for character recognition: comparison with gradient feature," in *International Conference on Document Analysis and Recognition*. IEEE, 2005, pp. 121–125.

[198] K. Ding, Z. Liu, L. Jin, and X. Zhu, "A comparative study of Gabor feature and gradient feature for handwritten chinese character recognition," in *International Conference on Wavelet Analysis and Pattern Recognition*. IEEE, 2007, pp. 1182–1186.

[199] H. E. S. Said, T. N. Tan, and K. D. Baker, "Personal identification based on handwriting," *Pattern Recognition*, pp. 149–160, 2000.

[200] A. Busch, W. W. Boles, and S. Sridharan, "Texture for script identification," *Pattern Analysis and Machine Intelligence*, pp. 1720–1732, 2005.

[201] Z. He, Y. Y. Tang, and X. You, "A contourlet-based method for writer identification," *Systems Man and Cybernetics*, pp. 364–368, 2005.

[202] F. Shahabi and M. Rahmati, "A new method for writer identification of handwritten Farsi documents," in *International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 426–430.

[203] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Texture-based descriptors for writer identification and verification," *Expert Systems with Applications*, pp. 2069–2080, 2013.

[204] A. Nicolaou, M. Liwicki, and R. Ingolf, "Oriented local binary patterns for writer identification," in *International Workshop on Automated Forensic Handwriting Analysis*, 2013.

[205] A. Nicolaou, F. Slimane, V. Märgner, and M. Liwicki, "Local binary patterns for Arabic optical font recognition," in *International Workshop on Document Analysis Systems*. IEEE, 2014, pp. 76–80.

[206] W. Jiang, A. T. Ho, H. Treharne, and Y. Q. Shi, "Local binary patterns for printer identification based on texture analysis," University of Surrey, Department of Computing, Tech. Rep. CS-11-05, September 2011.

[207] T. Furukawa, "A new method for discriminating printers based on contours qualities of printed characters using wavelet decomposition," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1115–1119.

[208] M. N. Maatouk, O. Jedidi, and N. E. B. Amara, "Watermarking ancient documents based on wavelet packets," in *Document Recognition and Retrieval*. SPIE, 2009.

[209] J. Liang, D. DeMenthon, and D. Doermann, "Geometric rectification of camera-captured document images." *Pattern Analysis and Machine Intelligence*, pp. 591–605, 2008.

[210] Y. Tian and S. G. Narasimhan, "Rectification and 3D reconstruction of curved document images," in *Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 377–384.

[211] C. A. B. Mello and R. D. Lins, "Generating paper texture of historical documents using statistical moments," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2000, pp. 1520–6149.

[212] P. Gupta, N. Vohra, S. Chaudhury, and S. D. Joshi, "Wavelet based page segmentation," in *Indian Conference on Computer Vision, Graphics and Image Processing*. IEEE, 2000, pp. 51–56.

[213] J. F. Cullen, J. J. Hull, and P. Hart, "Document image database retrieval and browsing using texture analysis," in *International Conference on Document Analysis and Recognition*. IEEE, 1997, pp. 718–721.

[214] D. Keysers, F. Shafait, and T. M. Breuel, "Document image zone classification - a simple high-performance approach," in *International Conference on Computer Vision Theory and Applications*. INSTICC Press, 2007, pp. 44–51.

[215] A. Gordo, F. Perronnin, and E. Valveny, "Large-scale document image retrieval and classification with runlength histograms and binary embeddings," *Pattern Recognition*, pp. 1898–1905, 2013.

[216] R. Vieux and J. P. Domenger, "Hierarchical clustering model for pixel-based classification of document images," in *International Conference on Pattern Recognition*. IEEE, 2012, pp. 290–293.

[217] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi, "Text extraction and document image segmentation using matched wavelets and MRF model," *Image Processing*, pp. 2117–2128, 2007.

[218] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, pp. 1167–1186, 1991.

[219] J. Li and R. M. Gray, "Context-based multiscale classification of document images using wavelet coefficient distributions," *Image Processing*, pp. 1604–1616, 2000.

[220] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, pp. 29–44, 2001.

[221] M. Benjlaiel, R. Mullot, and A. M. Alimi, "Multi-oriented handwritten annotations extraction from scanned documents," in *International Workshop on Document Analysis Systems*. IEEE, 2014, pp. 126–130.

[222] R. Pardeshi, B. B. Chaudhuri, M. Hangarge, and K. Santosh, "Automatic handwritten Indian scripts identification," in *International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 375–380.

[223] M. Seuret, M. Liwicki, and R. Ingold, "Pixel level handwritten and printed content discrimination in scanned documents," in *International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 423–428.

[224] D. Tao, L. Jin, S. Zhang, Z. Yang, and Y. Wang, "Sparse discriminative information preservation for Chinese character font categorization," *Neurocomputing*, pp. 159–167, 2014.

[225] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "ICDAR 2013 Competition on Historical Book Recognition (HBR 2013)," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1459–1463.

[226] ——, "ICDAR 2013 Competition on Historical Newspaper Layout Analysis (HNLA 2013)," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1454–1458.

[227] A. Crasson and J. D. Fekete, "Structuration des manuscrits : du corpus à la région," in *Colloque International Francophone sur l'Ecrit et le Document*, 2004.

[228] V. C. Kieu, M. Mehri, V. Rabeux, N. Journet, and M. Visani, "Génération d'images semi-synthétiques de documents anciens à des fins d'évaluation de performances et d'apprentissage," in *Colloque International Francophone sur l'Ecrit et le Document*, 2014, pp. 199–214.

[229] D. Coppi, C. Grana, and R. Cucchiara, "Illustrations segmentation in digitized documents using local correlation features," *Procedia Computer Science*, pp. 76–83, 2014.

[230] A. Garz and R. Sablatnig, "Multi-scale texture-based text recognition in ancient manuscripts," in *International Conference on Virtual Systems and Multimedia*. IEEE, 2010, pp. 336–339.

[231] H. Chouaib, F. Cloppet, and N. Vincent, "Graphical drop caps indexing," in *International Conference on Graphics recognition: achievements, challenges, and evolution*. Springer-Verlag, 2009, pp. 212–219.

[232] C. A. B. Mello, "Image segmentation of historical documents: using a quality index," in *International Conference on Image Analysis and Recognition*. Image Analysis and Recognition, Lecture Notes in Computer Science, 2004, pp. 209–216.

[233] C. A. B. Mello and R. D. Lins, "Image segmentation of historical documents," in *Visual2000*, 2000.

[234] T. K. Bhowmik and M. Kar, "Text localization in historical document images with local binary patterns and variance models," *PReMI*, pp. 501–508, 2013.

[235] R. Pareti and N. Vincent, "Ancient initial letters indexing," in *International Conference on Pattern Recognition*. IEEE, 2006, pp. 756–759.

[236] N. Zaghden, R. Mullot, and M. A. Alimi, "Categorizing ancient documents," *International Journal of Computer Science*, pp. 63–72, 2013.

[237] C. S. Ribeiro, J. M. Gil, J. R. C. Pinto, and J. M. da Costa Sousa, "Ancient document recognition using fuzzy methods," in *International Conference on Fuzzy Systems*. IEEE, 2005, pp. 833–838.

[238] M. A. Charrada and N. E. B. Amara, "Texture approach for nets extraction application to old Arab newspapers images structuring," in *Image Processing Theory, Tools and Applications*. IEEE, 2012, pp. 212–216.

[239] G. Zhong and M. Cheriet, "Image patches analysis for text block identification," in *International Conference on Information Science, Signal Processing and their Applications*. IEEE, 2012, pp. 1241–1246.

[240] A. Asi, R. Cohen, K. Kedem, J. El-Sana, and I. Dinstein, "A coarse-to-fine approach for layout analysis of ancient manuscripts," in *International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 140–145.

[241] G. Joutel, V. Eglin, S. Bres, and H. Emptoz, "Curvelets based feature extraction of handwritten shapes for ancient manuscripts classification," in *Document Recognition and Retrieval*. SPIE, 2007.

[242] A. Kricha and N. E. B. Amara, "Exploring textural analysis for historical documents characterization," *Journal of computing*, pp. 24–30, 2011.

[243] M. Benjelil, S. Kanoun, R. Mullot, and A. M. Alimi, "Complex documents images segmentation based on steerable pyramid features," *International Journal of Document Analysis and Recognition*, pp. 209–228, 2010.

[244] W. Boussellaa, A. Zahour, B. Taconet, A. Benabdelhafid, and A. Alimi, "Segmentation texte/graphique : application aux manuscrits Arabes anciens," in *Colloque International Francophone sur l'Ecrit et le Document*, 2006.

[245] C. Grana, D. Borghesani, and R. Cucchiara, "Automatic segmentation of digitalized historical manuscripts," *Multimedia Tools and Applications*, pp. 483–506, 2011.

[246] D. Hebert, T. Paquet, and S. Nicolas, "Continuous CRF with multi-scale quantization feature functions application to structure extraction in old newspaper," in *International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 493–497.

[247] A. Antonacopoulos and R. T. Ritchings, "Representation and classification of complex-shaped printed regions using white tiles," in *International Conference on Document Analysis and Recognition*. IEEE, 1995, pp. 1132–1135.

[248] A. K. Jain and Y. Zhong, "Page segmentation using texture analysis," *Pattern Recognition*, pp. 743–770, 1996.

[249] K. Etemad, D. Doermann, and R. Chellappa, "Multiscale segmentation of unstructured document pages using soft decision integration," *Pattern Analysis and Machine Intelligence*, pp. 92–96, 1997.

[250] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *Systems Man and Cybernetics*, pp. 269–285, 1976.

[251] C. H. Chen, "On the statistical image segmentation techniques," in *IEEE Conference on Pattern Recognition and Image Processing*. IEEE, 1981, pp. 262–266.

[252] J. M. H. DuBuf, M. Kardan, and M. Spann, "Texture feature performance for image segmentation," *Pattern Recognition*, pp. 291–309, 1990.

[253] S. W. Myint, N. S. N. Lam, and J. M. Tyler, "Wavelets for urban spatial feature discrimination: comparisons with fractal, spatial autocorrelation, and spatial co-occurrence approaches," *Photogrammetric Engineering & Remote Sensing*, pp. 803–812, 2004.

[254] K. I. Chang, K. W. Bowyer, and M. Sivagurunath, "Evaluation of texture segmentation algorithms," in *Computer Vision and Pattern Recognition*. IEEE, 1999, pp. 294–299.

[255] K. Baâti, S. Kanoun, and M. Benjlaiel, "Différenciation d'écritures Arabe et Latine de natures imprimée et manuscrite par approche globale," in *Colloque International Francophone sur l'Ecrit et le Document*, 2010.

[256] F. K. Jaiem, S. Kanoun, and V. Eglin, "Arabic font recognition based on a texture analysis," in *International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 673–677.

[257] H. Ma and D. Doermann, "Gabor filter based multi-class classifier for scanned document images," in *International Conference on Document Analysis and Recognition*. IEEE, 2003, pp. 968–972.

[258] K. Mouats, N. Journet, and R. Mullot, "Segmentation floue d'images de documents anciens par approche texture utilisant le filtre de Gabor," in *International Conference on Image and Signal Processing*, 2006.

[259] K. Mouats, "Segmentation d'images de documents anciens par approche texture - application du filtre de Gabor," Master's thesis, University of La Rochelle, La Rochelle, France, 2006.

[260] L. Wang and D. C. He, "Texture classification using texture spectrum," *Pattern Recognition*, pp. 905–910, 1990.

[261] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *International Conference on Pattern Recognition*. IEEE, 1994, pp. 582–585.

[262] D. Harwood, T. Ojala, M. Pietikäinen, S. Kelman, and L. Davis, "Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions," *Pattern Recognition Letters*, pp. 971–987, 1995.

[263] T. Ojala and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions," *Pattern Recognition*, pp. 477–486, 1999.

[264] L. Dua, X. Youa, H. Xua, Z. Gaoa, and Y. Tangb, "Wavelet domain local binary pattern features for writer identification," in *International Conference on Pattern Recognition*. IEEE, 2010, pp. 3691–3694.

[265] M. Lutf, X. You, and H. Li, "Offline Arabic handwriting identification using language diacritics," in *International Conference on Pattern Recognition*. IEEE, 2010, pp. 1912–1915.

[266] M. A. Ferrer, A. Morales, and U. Pal, "LBP based line-wise script identification," in *International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 369–373.

[267] X. Tang, "Texture information in run-length matrices," *Image Processing*, p. IEEE, 1998.

[268] R. W. Conners and C. A. Harlow, "A theoretical comparison of texture algorithms," *Pattern Analysis and Machine Intelligence*, pp. 204–222, 1980.

[269] N. Stamatopoulos, B. Gatos, and T. Georgiou, "Page frame detection for double page document images," in *International Workshop on Document Analysis Systems*, 2010, pp. 401–408.

[270] I. Dinstein and Y. Shapira, "Ancient Hebraic handwriting identification with run-length histograms," *Systems Man and Cybernetics*, pp. 405–409, 1982.

[271] M. Bulacu and L. Schomaker, "Automatic handwriting identification on medieval documents," in *International Conference on Image Analysis and Processing*, 2007, pp. 279–284.

[272] A. Ouji, Y. Leydier, and F. LeBourgeois, "Chromatic / achromatic separation in noisy document images," in *International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 167–171.

[273] S. Bres, "Contributions à la quantification des critères de transparence et d'anisotropie par une approche globale : application au contrôle de qualité de matériaux composites," Ph.D. dissertation, Institut National des Sciences Appliquées de Lyon, Lyon, France, 1994.

[274] A. K. Mikkilineni, P. J. Chiang, G. N. Ali, G. T. C. Chiu, J. P. Allebach, and E. J. D. III, "Printer identification based on graylevel co-occurrence features for security and forensic applications," in *Security, Steganography, and Watermarking of Multimedia Contents VII*. SPIE, 2005, pp. 430–440.

[275] M. Lin, J. Tapamo, and B. Ndovie, "A texture-based method for document segmentation and classification," *South African Computer Journal*, pp. 49–56, 2006.

[276] G. Peake and T. Tan, "Script and language identification from document images," in *Document Image Analysis*. IEEE, 1997, pp. 10–17.

[277] D. Gabor, "Theory of communication. Part 1: The analysis of information," *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, pp. 429–441, 1946.

[278] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *The Journal of Physiology*, pp. 551–566, 1968.

[279] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, pp. 1160–1169, 1985.

[280] Y. Zhu, T. Tan, and Y. Wang, "Biometric personal identification based on handwriting," in *International Conference on Pattern Recognition*. IEEE, 2000, pp. 797–800.

[281] J. Chen, H. Cao, R. Prasad, A. Bhardwaj, and P. Natarajan, "Gabor features for offline Arabic handwriting recognition," in *International Workshop on Document Analysis Systems*. ACM, 2010, pp. 53–58.

[282] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," *Pattern Recognition Letters*, pp. 2080–2092, 2005.

[283] R. Buse, Z. Q. Liu, and T. Caelli, "A structural and relational approach to handwritten word recognition," *Systems Man and Cybernetics*, pp. 847–861, 1997.

[284] X. Wang, X. Ding, and C. Liu, "Gabor filters-based feature extraction for character recognition," *Pattern Recognition*, pp. 369–379, 2005.

[285] Y. Zhu, T. Tan, and Y. Wang, "Font recognition based on global texture analysis," *Pattern Analysis and Machine Intelligence*, pp. 1192–1200, 2001.

[286] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification," *Pattern Analysis and Machine Intelligence*, pp. 751–756, 1998.

[287] G. D. Joshi, S. Garg, and J. Sivaswamy, "Script identification from Indian documents," in *International Workshop on Document Analysis Systems*. Springer-Verlag, 2006, pp. 255–267.

[288] H. B. Kekre and V. A. Bharadi, "Gabor filter based feature vector for dynamic signature recognition," *International Journal of Computer Applications*, pp. 74–80, 2010.

[289] M. Mu and Q. Ruan, "Mean and standard deviation as features for palmprint recognition based on Gabor filters," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 491–512, 2011.

[290] A. Sehad, Y. Chibani, and M. Cheriet, "Gabor filters for degraded document image binarization," in *International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 702–707.

[291] W. Chan and G. Coghill, "Text analysis using local energy," *Pattern Recognition*, pp. 2523–2532, 2001.

[292] S. S. Raju, P. B. Pati, and A. G. Ramakrishnan, "Gabor filters for document analysis in Indian bilingual documents," in *International Workshop on Document Image Analysis for Libraries*. IEEE, 2004, pp. 233–243.

[293] T. Randen and J. H. Husøy, "Segmentation of text/image documents using texture approaches," 1994.

[294] P. B. Pati, S. S. Raju, N. Pati, and A. G. Ramakrishnan, "Gabor filters for document analysis in Indian bilingual documents," in *International Conference on Intelligent Sensing and Information Processing*. IEEE, 2004, pp. 123–126.

[295] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *Pattern Analysis and Machine Intelligence*, pp. 55–73, 1990.

[296] A. Kricha, A. G. Lasmar, and N. E. B. Amara, "Exploration des ondelettes en prétraitement des documents anciens," in *Colloque International Francophone sur l'Ecrit et le Document*, 2006.

[297] C. W. Liang and P. Y. Chen, "DWT based text localization," *International Journal of Applied Science and Engineering*, pp. 105–116, 2004.

[298] P. S. Hiremath and S. Shivashankar, "Wavelet based co-occurrence histogram features for texture classification with an application to script identification in a document image," *Pattern Recognition Letters*, pp. 1182–1189, 2008.

[299] R. Manthalkar, P. K. Biswas, and B. N. Chatterji, "Rotation and scale invariant texture features using discrete wavelet packet transform," *Pattern Recognition Letters*, pp. 2455–2462, 2003.

[300] N. E. B. Amara and S. Gazzah, "Une approche d'identification des fontes Arabes," in *Colloque International Francophone sur l'Ecrit et le Document*, 2004.

[301] N. Zaghden, S. B. Moussa, and A. M. Alimi, "Reconnaissance des fontes Arabes par l'utilisation des dimensions fractales et des ondelettes," in *Colloque International Francophone sur l'Ecrit et le Document*, 2006.

[302] N. E. B. Amara and S. Gazzah, "Une approche a priori pour l'identification du scripteur en reconnaissance optique de l'écriture Arabe," in *Colloque International Francophone sur l'Ecrit et le Document*, 2006.

[303] N. B. Amor and N. E. B. Amara, "Analyse texturale de l'écriture arabe multifonte de gabor aux contourlets," in *Colloque International Francophone sur l'Ecrit et le Document*, 2008.

[304] S. Gazzah and N. E. B. Amara, "Arabic handwriting texture analysis for writer identification using the DWT-lifting scheme," in *International Conference on Document Analysis and Recognition*. IEEE, 2007, pp. 1133–1137.

[305] Z. He, B. Fang, J. Du, Y. Y. Tang, and X. You, "A novel method for offline handwriting-based writer identification," in *International Conference on Document Analysis and Recognition*. IEEE, 2005, pp. 242–246.

[306] X. Ding, L. Chen, and T. Wu, "Character independent font recognition on a single Chinese character," *Pattern Analysis and Machine Intelligence*, pp. 195–204, 2007.

[307] L. Zhang, Y. Lu, and C. L. Tan, "Italic font recognition using stroke pattern analysis on wavelet decomposed word images," in *International Conference on Pattern Recognition*. IEEE, 2004, pp. 835–838.

[308] S. A. Angadi and M. M. Kodabagi, "A fuzzy approach for word level script identification of text in low resolution display board images using wavelet features," in *International Conference on Advances in Computing, Communications and Informatics*. IEEE, 2013, pp. 1804–1811.

[309] N. B. Amor and N. E. B. Amara, "An approach for multifont Arabic characters features extraction based on contourlet transform," in *International Conference on Document Analysis and Recognition*. IEEE, 2007, pp. 1048–1052.

[310] S. Kumar, N. Khanna, S. Chaudhury, and S. D. Joshi, "Locating text in images using matched wavelets," in *International Conference on Document Analysis and Recognition*. IEEE, 2005, pp. 595–599.

[311] M. Acharyya and M. K. Kundu, "Document image segmentation using wavelet scale-space features," *Circuits and Systems for Video Technology*, pp. 1117–1127, 2002.

[312] N. Jin and Y. Y. Tang, "Text area localization under complex-background using wavelet decomposition," in *International Conference on Document Analysis and Recognition*. IEEE, 2001, pp. 1126–1130.

[313] Y. Y. Tang, H. Ma, J. Liu, B. F. Li, and D. Xi, "Multiresolution analysis in extraction of reference lines from documents with gray level background," *Pattern Analysis and Machine Intelligence*, pp. 921–926, 1997.

[314] S. Deivalakshmi, P. Palanisamy, and G. Vishwanathan, "A novel method for text and non-text segmentation in document images," in *International Conference on Communications and Signal Processing*. IEEE, 2013, pp. 255–259.

[315] L. Xavier, B. M. I. Thusnavis, and D. R. W. Newton, "Content based image retrieval using textural features based on pyramid-structure wavelet transform," in *International Conference on Electronics Computer Technology*. IEEE, 2011, pp. 79–83.

[316] H. S. Baird, M. A. Moll, C. An, and M. R. Casey, "Document image content inventories," in *Document Recognition and Retrieval*. SPIE, 2007.

[317] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, pp. 225–236, 2000.

[318] A. M. Vil'kin, I. V. Safonov, and M. A. Egorova, "Algorithm for segmentation of documents based on texture features," *Pattern Recognition and Image Analysis*, pp. 153–159, 2013.

[319] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *Systems Man and Cybernetics*, pp. 420–433, 1976.

[320] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man, and Cybernetics*, pp. 62–66, 1979.

[321] L. Shijian and C. L. Tan, "Script and language identification in noisy and degraded document images," *Pattern Analysis and Machine Intelligence*, pp. 14–24, 2008.

[322] J. He, Q. D. M. Do, A. C. Downton, and J. H. Kim, "A comparison of binarization methods for historical archive documents," in *International Conference on Document Analysis and Recognition*. IEEE, 2005, pp. 538–542.

[323] A. G. Lasmar, A. Kricha, and N. E. B. Amara, "A segmentation text/background method for degraded ancient Arabic manuscript," in *International Conference on Information & Communication Technologies*. IEEE, 2006, pp. 1327–1331.

[324] L. Cinque, L. Lombardi, and G. Manzini, "A multiresolution approach for page segmentation," *Pattern Recognition Letters*, pp. 217–225, 1998.

[325] C. Tan and P. Ng, "Text extraction using pyramid," *Pattern Recognition*, pp. 63–72, 1998.

[326] C. Tan and Z. Zhang, "Text block segmentation using pyramid structure," in *Document Recognition and Retrieval*. SPIE, 2000, pp. 297–306.

[327] A. Lemaitre, J. Camillerapp, and B.Coüasnon, "Multiresolution cooperation improves document structure recognition," *International Journal of Document Analysis and Recognition*, pp. 97–109, 2008.

[328] H. Greenspan, "Multi-resolution image processing and learning for texture recognition and image enhancement," Ph.D. dissertation, California Institute of Technology, California, USA, 1994.

[329] S. Contassot-Vivier, G. L. Bosco, and N. C. Dao, "Multiresolution approach for image processing," in *Erasmus ICP-A-2007*, 1996.

[330] G. Nguyen, M. Coustaty, and J. M. Ogier, "Stroke feature extraction for lettrine indexing," in *International Conference on Image Processing Theory Tools and Applications*. IEEE, 2010, pp. 355–360.

[331] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967, pp. 281–297.

[332] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies 1. Hierarchical systems," *The Computer Journal*, pp. 373–380, 1967.

[333] F. Lalys, C. Haegelen, M. Mehri, S. Drapier, M. Vérin, and P. Jannin, "Anatomo-clinical atlases correlate clinical data and electrode contact coordinates : application to subthalamic deep brain stimulation," *Journal of Neuroscience*, pp. 297–307, 2013.

[334] H. P. Lai, M. Visani, A. Boucher, and J. M. Ogier, "An experimental comparison of clustering methods for content-based indexing of large image databases," *Pattern Analysis and Applications*, pp. 345–366, 2012.

[335] J. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, pp. 236–244, 1963.

[336] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher, "A realistic dataset for performance evaluation of document layout analysis," in *International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 296–300.

[337] D. Doermann, E. Zotkina, and H. Li, "GEDI - a groundtruthing environment for document images," in *International Workshop on Document Analysis Systems*. ACM, 2010.

[338] B. A. Yanikoglu and L. Vincent, "Pink Panther: a complete environment for ground-truthing and benchmarking document page," *Pattern Recognition*, pp. 1191–1204, 1998.

[339] A. Silva, "Metrics for evaluating performance in document analysis: application to tables," *International Journal of Document Analysis and Recognition*, pp. 101–109, 2011.

[340] F. Ge, S.Wang, and T. Liu, "New benchmark for image segmentation evaluation," *Journal of Electronic Imaging*, pp. 1–16, 2007.

[341] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, pp. 53–65, 1987.

[342] P. C. Saxena and K. Navaneetham, "The effect of cluster size, dimensionality, and number of clusters on recovery of true cluster structure through Chernoff-type faces," *Journal of the Royal Statistical Society, The Statistician*, pp. 415–425, 1991.

[343] J. R. Jensen, *Introductory digital image processing.* Prentice-Hall, Englewood Cliffs, NJ, 1986.

[344] P. M. Mather, *Computer processing of remotely-sensed images: an introduction.* 2nd Edition John Wiley & Sons, 1999.

[345] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *DARPA Broadcast News Workshop.* Morgan Kaufmann Publishers, Inc, 1999, pp. 249–252.

[346] J. M. Wei, X. J. Yuan, Q. H. Hub, and S. Q. Wangc, "A novel measure for evaluating classifiers," *Expert Systems with Applications*, pp. 3799–3809, 2010.

[347] D. M. W. Powers, "Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, pp. 37–63, 2011.

[348] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data.* Springer-Verlag, 2011.

[349] A. K. Santra and C. J. Christy, "Genetic algorithm and confusion matrix for document clustering," *International Journal of Computer Science*, pp. 322–328, 2012.

[350] H. Abedi, H. Rostami, and S. Rahimi, "A hybrid approach for optimal feature selection based on evolutionary algorithms and classic approaches," *Advances in Computer Science: an International Journal*, 2013.

[351] C. An and H. S. Baird, "The convergence of iterated classification," in *International Workshop on Document Analysis Systems.* IEEE, 2008, pp. 663–670.

[352] J. Cocquerez and S. Philipp, *Analyse d'images : filtrage et segmentation.* Masson, 1995.

[353] R. Duda, P. Hart, and D. Stork, *Pattern classification.* 2nd Edition Wiley-Interscience, 2001.

[354] M. Cord and P. Cunningham, *Machine learning techniques for multimedia case studies on organization and retrieval, series: cognitive technologies.* Springer-Verlag, 2008.

[355] A. Cornuéjols and L. Miclet, *Apprentissage artificiel : concepts et algorithmes.* 2nd Edition Eyrolles, 2010.

[356] D. J. Ketchen and C. L. Shook, "The application of cluster analysis in strategic management research: an analysis and critique," *Strategic Management Journal*, pp. 441–458, 1996.

[357] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pp. 411–423, 2001.

[358] C. Goutte, L. K. Hansen, M. G. Liptrot, and E. Rostrup, "Feature-space clustering for fMRI meta-analysis," *Human Brain Mapping*, pp. 165–183, 2001.

[359] H. Akaike, "A new look at the statistical model identification," *Automatic Control*, pp. 716–723, 1974.

[360] G. Schwartz, "Estimating the dimension of a model," in *The Annals of Statistics.* Institute of Mathematical Statistics, 1978, pp. 461–464.

[361] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *Pattern Analysis and Machine Intelligence*, pp. 719–725, 2000.

[362] J. F. Hair, R. Anderson, R. Tatham, and W. C. Black, *Multivariate data analysis.* Prentice Hall, 1992.

[363] C. A. Sugar and G. M. James, "Finding the number of clusters in a data set: an information theoretic approach," *Journal of the American Statistical Association*, pp. 750–763, 2003.

[364] F. Can and E. A. Ozkarahan, "Concepts and effectiveness of the clustering methodology for text databases," *ACM Transactions on Database Systems*, pp. 483–517, 1990.

[365] M. Honarkhah and J. Caers, "Stochastic simulation of patterns using distance-based pattern modeling," *Mathematical Geosciences*, pp. 487–517, 2010.

[366] M. Schonlau, "The clustergram: a graph for visualizing hierarchical and nonhierarchical cluster analyses," *The Stata Journal*, pp. 391–402, 2002.

[367] N. Iam-on and S. Garrett, "LinkCluE: a Matlab package for link-based cluster ensembles," *Journal of Statistical Software*, pp. 1–36, 2010.

[368] W. S. Sarle, "The cubic clustering criterion," SAS Institute, Tech. Rep. SAS Technical Report A-108: The Cubic Clustering Criterion, 1983.

[369] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in color image segmentation," in *International Conference on Advances in Pattern Recognition and Digital Techniques.* Narosa Publishing House, 1999, pp. 137–143.

[370] H. A. Moesa, D. B. K.C., and T. Akutsu, "Efficient determination of cluster boundaries for analysis of gene expression profile data using hierarchical clustering and wavelet transform," *Genome Informatics*, pp. 132–141, 2005.

[371] R. Lletía, M. C. Ortiza, L. A. Sarabiab, and M. S. Sánchez, "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes," in *Colloquim Chemiometricum Mediterraneum.* Elsevier Science, Analytica Chimica Acta, 2004, pp. 87–100.

[372] StatSoft. (2010) Finding the right number of clusters in k-means and EM clustering: v-Fold Cross-Validation. Electronic Statistics Textbook. [Online]. Available: http://www.statsoft.com/textbook/cluster-analysis/

[373] Q. Zhao, M. Xu, and P. Fränti, "Extending external validity measures for determining the number of clusters," in *International Conference on Intelligent Systems Design and Applications.* IEEE, 2011, pp. 931–936.

[374] K. Kryszczuk and P. Hurley, "Estimation of the number of clusters using multiple clustering validity indices," in *International Conference on Multiple Classifier Systems.* Springer-Verlag, 2010, pp. 114–123.

[375] N. Bolshakova and F. Azuaje, "Estimating the number of clusters in DNA microarray data," *Methods of information in medicine*, pp. 153–157, 2006.

[376] G. B. Mufti, P. Bertrand, and L. E. Moubarki, "Determining the number of groups from measures of cluster stability," *Applied Stochastic Models and Data Analysis*, pp. 404–413, 2005.

[377] T. Simpson, J. Armstrong, and A. Jarman, "Merged consensus clustering to assess and improve class discovery with microarray data," *Boston Medical Center Bioinformatics*, pp. 1471–1482, 2010.

[378] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus Clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, pp. 91–118, 2003.

[379] D. E. Knuth, *The art of computer programming, volume 3: (2nd ed.) sorting and searching.* Addison Wesley Longman Publishing Co, 1997.

[380] P. Mahalanobis, "On the generalised distance in statistics," in *Proceedings of the National Institute of Sciences of India.* NISI, 1936, pp. 49–55.

[381] A. K. Jain and R. C. Dubes, *Algorithms for clustering data.* Prentice-Hall, 1998.

[382] W. J. Krzanowski and Y. T. Lai, "A criterion for determining the number of groups in a data set using sum-of-squares clustering," *International Biometric Society, JSTOR*, pp. 23–34, 1988.

[383] J. A. Hartigan, *Clustering algorithms.* John Wiley & Sons, 1975.

[384] R. B. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, pp. 1–27, 1974.

[385] A. J. Scott and M. J. Symons, "Clustering methods based on likelihood ratio criteria," *Biometrics*, pp. 387–397, 1971.

[386] F. H. Marriott, "Practical problems in a method of cluster analysis," *Biometrics*, pp. 501–514, 1971.

[387] G. W. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, pp. 159–179, 1985.

[388] H. P. Friedman and J. Rubin, "On some invariant criteria for grouping data," *Journal of the American Statistical Association*, pp. 1159–1178, 1967.

[389] J. Rubin, "Optimal classification into groups: an approach for solving the taxonomy problem," *Journal of Theoretical Biology*, pp. 103–144, 1967.

[390] L. J. Hubert and J. R. Levin, "A general statistical framework for assessing categorical clustering in free recall," *Psychological Bulletin*, pp. 1072–1080, 1976.

[391] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *Pattern Analysis and Machine Intelligence*, pp. 224–227, 1979.

[392] D. A. Ratkowsky and G. N. Lance, "A criterion for determining the number of groups in a classification," *Australian Computer Journal*, pp. 115–117, 1978.

[393] G. H. Ball and D. J. Hall, "ISODATA, a novel method of data analysis and pattern classification," Menlo Park: Stanford Research Institute, Tech. Rep. AD0699616, 1965.

[394] G. W. Milligan, "An examination of the effect of six types of error perturbation on fifteen clustering algorithms," *Psychometrika*, pp. 325–342, 1980.

[395] T. Frey and H. V. Groenewoud, "A cluster analysis of the d-squared matrix of white spruce stands in saskatchewan based on the maximum-minimum principle," *Journal of Ecology*, pp. 873–886, 1972.

[396] J. O. McClain and V. R. Rao, "CLUSTISZ: a program to test for the quality of clustering of a set of objects," *Journal of Marketing Research*, pp. 456–460, 1975.

[397] J. Dunn, "Well separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, pp. 95–104, 1974.

[398] M. Halkidi, M. Vazirgiannis, and I. Batistakis, "Quality scheme assessment in the clustering process," in *Principles and Practice of Knowledge in databases.* Springer-Verlag, 2000, pp. 265–276.

[399] M. Halkidi, I. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, pp. 107–145, 2001.

[400] E. Deza and M. M. Deza, *Encyclopedia of distances.* Springer-Verlag, 2013.

[401] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, pp. 846–850, 1971.

[402] L. Hubert and P. Arabic, "Comparing partitions," *Journal of Classification*, pp. 193–218, 1985.

[403] A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger, "Hierarchical clustering based on mutual information," in *Quantitative Methods (q-bio.QM).* CoRR q-bio.QM/0311039, 2003, pp. 193–218.

[404] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, pp. 2837–2854, 2010.

[405] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American Statistical Association*, pp. 553–569, 1983.

[406] T. Chang and C. C. J. Kuo, "Texture segmentation with tree-structured wavelet transform," in *International Symposium on Time-Frequency and Time-Scale Analysis.* IEEE, 1992, pp. 543–546.

[407] K. Etemad, D. Doermann, and R. Chellappa, "Page segmentation using decision integration and wavelet packets," in *International Conference on Pattern Recognition.* IEEE, 1994, pp. 345–349.

[408] T. Palfray, D. H. P. Tranouez, S. Nicolas, and T. Paquet, "Segmentation logique d'images de journaux anciens," in *Colloque International Francophone sur l'Ecrit et le Document*, 2012.

[409] M. Rais, N. A. Goussies, and M. Mejail, "Using adaptive run-length smoothing algorithm for accurate text localization in images," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science.* Springer-Verlag, 2011, pp. 149–156.

[410] B. Gatos, S. L. Mantzaris, S. J. Perantonis, and A. Tsigris, "Automatic page analysis for the creation of a digital library from newspaper archives," *International Journal on Digital Libraries*, pp. 77–84, 2000.

[411] N. Papamarkos, J. Tzortzakis, and B. Gatos, "Determination of run-length smoothing values for document segmentation," in *International Conference on Electronics, Circuits, and Systems.* IEEE, 1996, pp. 684–687.

[412] H. M. Sun, "Page segmentation for Manhattan and non-Manhattan layout documents via selective CRLA," in *International Conference on Document Analysis and Recognition.* IEEE, 2005, pp. 116–120.

[413] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Addison-Wesley, 1992.

[414] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis, "Keyword-guided word spotting in historical printed documents using synthetic data and user feedback," *International Journal of Document Analysis and Recognition*, pp. 167–177, 2007.

[415] S. Ferilli, F. Leuzzi, F. Rotella, and F. Esposito, "A run-length smoothing-based algorithm for non-Manhattan document segmentation," in *Convegno del Gruppo Italiano Ricercatori in Pattern Recognition*, 2012.

[416] S. Ferilli, M. Biba, F. Esposito, and T. M. A. Basile, "A distance-based technique for non-Manhattan layout analysis," in *International Conference on Document Analysis and Recognition.* IEEE, 2009, pp. 231–235.

[417] S. Arora, D. Sharma, and S. Arora, "Document image segmentation using dynamic thresholds and identification of each region type," *International Journal of Information & Computation Technology*, pp. 1869–1875, 2014.

[418] V. G. Brunet, M. Manouvrier, and M. Rukoz, "Synthèse sur les modèles de représentation des relations spatiales dans les images symboliques," *Revue des Nouvelles Technologies de l'Information*, pp. 19–54, 2008.

[419] C. Hudelot, J. Atif, and I. Bloch, "Fuzzy spatial relation ontology for image interpretation," *Fuzzy Sets and Systems*, pp. 1929–1951, 2008.

[420] S. Y. Lee and F. J. Hsu, "Picture algebra for spatial reasoning of iconic images represented in 2-D C-string," *Pattern Recognition Letters*, pp. 425–435, 1991.

[421] Y. Niu, M. T. Ozsu, and X. Li, "Two-dimensional S-tree: an index structure for content-based retrieval of images," in *Multimedia Computing and Networking.* SPIE, 1999, pp. 110–121.

[422] G. Huang, W. Zhang, and L. Wenyin, "A discriminative representation for symbolic image similarity evaluation," in *International Workshop on Graphics Recognition.* Springer-Verlag, 2008, pp. 71–79.

[423] C. Freksa, "Temporal reasoning based on semi-intervals," *Artificial Intelligence*, pp. 199–227, 1992.

[424] W. H. Yeh and Y. I. Chang, "An efficient iconic indexing strategy for image rotation and reflection in image databases," *Journal of Systems and Software*, pp. 1184–1195, 2008.

[425] P. W. Huang and C. H. Lee, "Image database design based on 9D-SPA representation for spatial relations," in *Knowledge and Data Engineering.* IEEE, 2004, pp. 1486–1496.

[426] S. Y. Lee, M. C. Yang, and J. W. Chen, "Signature file as a spatial filter for iconic image database," *Journal of Visual Languages & Computing*, pp. 373–397, 1992.

[427] C. C. Chang and C. F. Lee, "A bin-tree oriented iconic indexing scheme for retrieving symbolic pictures," *Data & Knowledge Engineering*, pp. 121–133, 1998.

[428] M. J. Egenhofer and R. D. Franzosa, "Point-set topological spatial relations," *International Journal of Geographical Information Systems*, pp. 161–174, 1991.

[429] E. Clementini and R. Laurini, "Un cadre conceptuel pour modéliser les relations spatiales," *Revue des Nouvelles Technologies de l'Information*, pp. 1–17, 2008.

[430] L. P. D. L. Heras, "Relational models for visual understanding of graphical documents: application to architectural drawings," Ph.D. dissertation, Universitat Autònoma de Barcelona, Barcelona, Spain, 2014.

[431] M. Coustaty, A. Bouju, K. Bertet, and G. Louis, "Using ontologies to reduce the semantic gap between historians and image processing algorithms," in *International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 156–160.

[432] M. Coustaty, J. M. Ogier, and R. P. N. Vincent, "Drop caps decomposition for indexing: a new letter extraction method," in *International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 476–480.

[433] C. Hudelot, J. Atif, and I. Bloch, "FSRO : une ontologie de relations spatiales floues pour l'inteprétation d'images," *Revue des Nouvelles Technologies de l'Information*, pp. 53–84, 2008.

[434] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, pp. 11–32, 1991.

[435] D. O. Hebb, *The organization of behavior: a neuropsychological theory*. John Wiley & Sons, 1949.

[436] L. Goldfarb, "Representational formalisms: Why we haven't had one," in *Pattern representation and the future of pattern recognition: a program for action: ICPR satellite workshop*, 2004.

[437] L. Goldfarb and D. Gay, "What is a structural representation ?" Faculty of Computer Science, University of New Brunswick, Fredericton, Canada, Tech. Rep. TR05-175, 2006.

[438] K. M. Borgwardt, "Graph kernels," Ph.D. dissertation, University of Ludwig-Maximilians, Munich, Germany, 2007.

[439] K. Riesen, "Classification and clustering of vector space embedded graphs," Ph.D. dissertation, University of Bern, Bern, Switzerland, 2009.

[440] J. R. Ullmann, "An algorithm for sub-graph isomorphism," *Journal of the Association for Computing Machinery*, pp. 31–42, 1976.

[441] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 265–298, 2004.

[442] P. Mahé, N. Ueda, T. Akutsu, J. L. Perret, and J. P. Vert, "Graph kernels for molecular structure-activity relationship analysis with support vector machines," *Journal of Chemical Information and Modeling*, pp. 939–951, 2005.

[443] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H. P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, pp. 47–56, 2005.

[444] K. M. Borgwardt and H. P. Kriegel, "Shortest-path kernels on graphs," in *International Conference on Data Mining*. IEEE, 2005, pp. 74–81.

[445] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi, "Graph kernels for chemical informatics," *Neural Networks*, pp. 1093–1110, 2005.

[446] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," in *Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[447] B. L. Saux and H. Bunke, "Feature selection for graph-based image classifiers," in *Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*. Springer-Verlag, 2005, pp. 147–154.

[448] B. Luo, R. C. Wilson, and E. R. Hancock, "Spectral embedding of graphs," *Pattern Recognition*, pp. 2213–2223, 2003.

[449] C. C. Aggarwal and H. Wang, *Managing and mining graph data*. Database Management & Information Retrieval, Advances in Database Systems, Springer, 2010.

[450] D. Maio and D. Maltoni, "A structural approach to fingerprint classification," in *International Conference on Pattern Recognition*. IEEE, 1996, pp. 578–585.

[451] M. Neuhaus and H. Bunke, "An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification," in *Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science*. Springer-Verlag, 2004, pp. 180–189.

[452] S. Fischer, "Automatic identification of diatoms," Ph.D. dissertation, University of Bern, Bern, Switzerland, 2002.

[453] R. Ambauen, S. Fischer, and H. Bunke, "Graph edit distance with node splitting and merging, and its application to diatom identification," in *International Workshop on Graph Based Representations in Pattern Recognition*. Springer-Verlag, 2003, pp. 95–106.

[454] P. J. Dickinson, H. Bunke, A. Dadej, and M. Kraetzl, "On graphs with unique node labels," in *International Workshop on Graph Based Representations in Pattern Recognition*. Springer-Verlag, 2003, pp. 13–23.

[455] ——, "Matching graphs with unique node labels," *Pattern Analysis and Applications*, pp. 243–254, 2004.

[456] P. J. Dickinson, M. Kraetzl, H. Bunke, M. Neuhaus, and A. Dadej, "Similarity measures for hierarchical representations of graphs with unique node labels," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 425–442, 2004.

[457] H. N. Ho, C. Rigaud, J. C. Burie, and J. M. Ogier, "Detecting recurring deformable objects: an approximate graph matching method for detecting characters in comics books," in *International Workshop on Graphics Recognition*. Springer-Verlag, 2013, pp. 122–134.

[458] J. Lladós and G. Sánchez, "Graph matching versus graph parsing in graphics recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 455–475, 2004.

[459] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "Fast graph matching for detecting CAD image components," in *International Conference on Pattern Recognition*. IEEE, 2000, pp. 1034–1037.

[460] J. Rocha and T. Pavlidis, "A shape analysis model with applications to a character recognition system," *Pattern Analysis and Machine Intelligence*, pp. 393–404, 1994.

[461] P. N. Suganthan and H. Yan, "Graph kernels for chemical informatics," *Image and Vision Computing*, pp. 191–201, 1998.

[462] A. Schenker, "Graph-theoretic techniques for web content mining," Ph.D. dissertation, University of South Florida, Tampa, USA, 2003.

[463] A. Schenker, M. Last, H. Bunke, and A. Kandel, "Classification of Web documents using graph matching," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 475–496, 2004.

[464] A. Schenker, H. Bunke, M. Last, A. Kandel, and D. Schenker, *Graph-theoretic techniques for Web content mining*. World Scientific Series in Machine Perception and Artificial Intelligence, 2005.

[465] H. Bunke, "Recent advances in structural pattern recognition with applications to visual form analysis," in *Visual Form, Lecture Notes in Computer Science*. Springer-Verlag, 2001, pp. 11–23.

[466] S. Jouili and S. Tabbone, "Applications des graphes en traitement d'images," in *International Conference on Relations, Orders and Graph: Interaction with Computer Science*. Springer-Verlag, 2008, pp. 434–442.

[467] S. Jouili, S. Tabbone, and E. Valveny, "Comparing graph similarity measures for graphical recognition," in *International Workshop on Graphics Recognition*. Springer-Verlag, 2010, pp. 37–48.

[468] S. O. Belkasim, M. Shridhar, and M. Ahmadi, "Pattern recognition with moment invariants: a comparative study and new results," *Pattern Recognition*, pp. 1117–1138, 1991.

[469] S. Brunessaux, P. Giroux, B. Grilheres, M. Manta, M. Bodin, K. Choukri, O. Galibert, and J. Kahn, "The Maurdor project: improving automatic processing of digital documents," in *International Workshop on Document Analysis Systems*. IEEE, 2014, pp. 349–354.

[470] S. Mao and T. Kanungo, "Software architecture of PSET: a page segmentation evaluation toolkit," *International Journal of Document Analysis and Recognition*, pp. 205–217, 2002.

[471] C. Wolf and J. M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition*, pp. 280–296, 2006.

[472] H. Bunke and G. Allermann, "Inexact graph matching for structural pattern recognition," *Pattern Recognition Letters*, pp. 245–253, 1983.

[473] E. Gudes, S. Shimony, and N. Vanetik, "Discovering frequent graph patterns using disjoint paths," *Knowledge and Data Engineering*, pp. 1441–1456, 2006.

[474] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *The Knowledge Engineering Review*, pp. 75–105, 2013.

[475] Z. Zou, J. Li, H. Gao, and S. Zhang, "Mining frequent subgraph patterns from uncertain graph data," *Knowledge and Data Engineering*, pp. 1203–1218, 2010.

[476] Y. Chen, S. Sanghavi, and H. Xu, "Improved graph clustering," *Information Theory*, pp. 6440–6455, 2014.

[477] H. N. Djidjev and M. Onus, "Scalable and accurate graph clustering and community structure detection," *Parallel and Distributed Systems*, pp. 1022–1029, 2013.

[478] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *Cybernetics*, p. 1, 2014.

[479] K. Riesen and H. Bunke, "Graph classification by means of Lipschitz embedding," *Systems, Man, and Cybernetics Systems, Part B: Cybernetics,*, pp. 1472–1483, 2009.

[480] D. Justice and A. Hero, "A binary linear programming formulation of the graph edit distance," *Pattern Analysis and Machine Intelligence*, pp. 1200–1214, 2006.

[481] F. B. Silva, S. Tabbone, and R. D. S. Torres, "BoG: a new approach for graph matching," in *International Conference on Pattern Recognition*. IEEE, 2014, pp. 82–87.

[482] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman & Co, 1979.

[483] E. Bengoetxea, "Inexact graph matching using estimation of distribution algorithms," Ph.D. dissertation, École Nationale Supérieure des Télécommunications, Paris, France, 2002.

[484] P. Foggia and M. Vento, "Graph embedding for pattern recognition," in *Recognizing Patterns in Signals, Speech, Images and Videos, Lecture Notes in Computer Science*. Springer-Verlag, 2010, pp. 75–82.

[485] K. Riesen and H. Bunke, *Graph classification and clustering based on vector space embedding*. World Scientific Publishing Co, 2010.

[486] D. Raviv, R. Kimmel, and A. M. Bruckstein, "Graph isomorphisms and automorphisms via spectral signatures," *Pattern Analysis and Machine Intelligence*, pp. 1985–1993, 2013.

[487] J. S. Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[488] B. Schölkopf and A. J. Smola, *Learning with kernels*. MIT Press, 2002.

[489] T. Gärtner, "A survey of kernels for structured data," *SIGKDD Explorations*, pp. 49–58, 2003.

[490] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: hardness results and efficient alternatives," in *Learning Theory and Kernel Machines, Lecture Notes in Computer Science*. Springer-Verlag, 2003, pp. 129–143.

[491] C. Watkins, "Kernels from matching operations," Department of Computer Science, Royal Holloway, University of London, Tech. Rep. CSD-TR-98-07, 1999.

[492] K. M. Borgwardt, T. Petri, S. V. N. Vishwanathan, and H. P. Kriegel, "An efficient sampling scheme for comparison of large graphs," in *International Workshop on Mining and Learning with Graphs*. P. Frasconi, K. Kersting, and K. Tsuda, editors, 2007.

[493] R. I. Kondor and J. D. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc, 2002, pp. 315–322.

[494] J. Ramon and T. Gärtner, "Expressivity versus efficiency of graph kernels," in *International Workshop on Mining Graphs, Trees and Sequences*. T. Washio and L. D. Raedt, editors, 2003, pp. 65–74.

[495] T. Horvíth, T. Gärtner, and S. Augustin, "Cyclic pattern kernels for predictive graph mining," in *International conference on Knowledge discovery and data mining*. ACM, 2004, pp. 158–167.

[496] S. Kramer and L. D. Raedt, "Feature construction with version spaces for biochemical applications," in *International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc, 2001, pp. 258–265.

[497] E. Barbu, P. Héroux, S. Adam, and E. Trupin, "Clustering document images using a bag of symbols representation," in *International Conference on Document Analysis and Recognition*. IEEE, 2005, pp. 1216–1220.

[498] N. Sidère, P. Héroux, and J. Y. Ramel, "Vector representation of graphs: application to the classification of symbols and letters," in *International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 681–685.

[499] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence*, pp. 731–737, 1997.

[500] P. Ren, R. C. Wilson, and E. R. Hancock, "Graph characterization via Ihara coefficients," *Neural Networks*, pp. 233–245, 2010.

[501] M. Neuhaus and H. Bunke, *Bridging the gap between graph edit distance and kernel machines*. World Scientific, Series in Machine Perception and Artificial Intelligence, 2007.

[502] P. Foggia, G. Percannella, and M. Vento, "Graph matching and learning in pattern recognition in the last 10 years," *International Journal of Pattern Recognition and Artificial Intelligence*, 2014.

[503] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph," *Pattern Recognition Letters*, pp. 255–259, 1998.

[504] W. D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray, "Graph distances using graph union," *Pattern Recognition Letters*, pp. 701–704, 2001.

[505] M. L. Fernández and G. Valiente, "A graph distance metric combining maximum common subgraph and minimum common supergraph," *Pattern Recognition Letters*, pp. 753–758, 2001.

[506] S. Umeyama, "An eigen-decomposition approach to weighted graph matching problems," *Pattern Analysis and Machine Intelligence*, pp. 695–703, 1988.

[507] G. Zhao, B. Luo, J. Tang, and J. Ma, "Using eigen-decomposition method for weighted graph matching," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, Lecture Notes in Computer Science*. Springer-Verlag, 2007, pp. 1283–1294.

[508] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *Pattern Analysis and Machine Intelligence*, pp. 377–388, 1996.

[509] B. J. V. Wyk and M. A. V. Wyk, "A POCS-based graph matching algorithm," *Pattern Analysis and Machine Intelligence*, pp. 1526–1530, 2004.

[510] K. Riesen, M. Neuhaus, and H. Bunke, "Bipartite graph matching for computing the edit distance of graphs," in *Graph-Based Representations in Pattern Recognition, Lecture Notes in Computer Science*. Springer-Verlag, 2007, pp. 1–12.

[511] K. Riesen, S. Fankhauser, H. Bunke, and P. Dickinson, "Efficient suboptimal graph isomorphism," in *Graph-Based Representations in Pattern Recognition, Lecture Notes in Computer Science*. Springer-Verlag, 2009, pp. 1–12.

[512] S. Fankhauser, K. Riesen, H. Bunke, and P. J. Dickinson., "Suboptimal graph isomorphism using bipartite matching," *International Journal of Pattern Recognition and Artificial Intelligence*, 2012.

[513] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image and Vision Computing*, pp. 950–959, 2009.

[514] M. Neuhaus, K. Riesen, and H. Bunke, "Fast suboptimal algorithms for the computation of graph edit distance," in *Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science*. Springer-Verlag, 2006, pp. 163–172.

[515] R. Myers, R. C. Wison, and E. R. Hancock, "Bayesian graph edit distance," *Pattern Analysis and Machine Intelligence*, pp. 628–635, 2000.

[516] R. Raveaux, J. C. Burie, and J. M. Ogier, "A graph matching method and a graph matching distance based on subgraph assignments," *Pattern Recognition Letters*, pp. 394–406, 2010.

[517] K. Riesen, S. Fankhauser, and H. Bunke, "Speeding up graph edit distance computation with a bipartite heuristic," in *Mining and Learning with Graphs*. Springer-Verlag, 2007.

[518] S. Fankhauser, K. Riesen, and H. Bunke, "Speeding up graph edit distance computation through fast bipartite matching," in *Graph-Based Representations in Pattern Recognition, Lecture Notes in Computer Science*. Springer-Verlag, 2011, pp. 102–111.

[519] Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou, "Comparing stars: on approximating graph edit distance," in *Proceedings of the VLDB Endowment*. Springer-Verlag, 2009, pp. 25–36.

[520] L. Livi and A. Rizzi, "The graph matching problem," *Pattern Analysis and Applications*, pp. 253–283, 2013.

[521] X. Gao, B. Xiao, D. Tao, and X. Li, "A survey of graph edit distance," *Pattern Analysis and Applications*, pp. 113–129, 2010.

[522] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics*, pp. 100–107, 1968.

[523] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke, "Approximation of graph edit distance based on hausdorff matching," *Pattern Recognition*, pp. 331–343, 2015.

[524] H. A. Almohamad and S. O. Duffuaa, "A linear programming approach for the weighted graph matching problem," *Pattern Analysis and Machine Intelligence*, pp. 522–525, 1993.

[525] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, pp. 32–38, 1957.

[526] K. Riesen, A. Fischer, and H. Bunke, "Improving approximate graph edit distance using genetic algorithms," in *Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science*. Springer-Verlag, 2014, pp. 63–72.

[527] R. C. Wilson and E. R. Hancock, "Structural matching by discrete relaxation," *Pattern Analysis and Machine Intelligence*, pp. 634–648, 1997.

[528] J. Lerouge, P. LeBodic, P. Héroux, and S. Adam, "GEM++: a tool for solving substitution-tolerant subgraph isomorphism," in *International Workshop on Graph Based Representations in Pattern Recognition*. Springer-Verlag, to be published, 2015.

[529] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence*, pp. 1798–1828, 2013.

[530] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *International Conference on Computer Vision*. IEEE, 2009, pp. 670–677.

[531] R. Achanta, A. Shaji, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence*, pp. 2274–2282, 2012.

[532] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition*. IEEE, 2005, pp. 60–65.

[533] J. Darbon and M. Sigelle, "Image restoration with discrete constrained total variation part I: fast and exact optimization," *Journal of Mathematical Imaging and Vision*, pp. 261–276, 2006.

[534] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the Fuzzy C-Means clustering algorithm," in *Computers & Geosciences*. Pergamon Press, 1984, pp. 191–203.

[535] F. Kovács, C. Legány, and A. Babos, "Cluster validity measurement techniques," in *International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*. World Scientific and Engineering Academy and Society, 2006, pp. 388–393.

[536] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 226–231.

[537] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," in *International Conference on Management of Data*. ACM Press, 1999, pp. 49–60.

[538] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 1997.

[539] W. Wang, J. Yang, and R. Muntz, "STING: a statistical information grid approach to spatial data mining," in *International Conference on Very Large Data*. Morgan Kaufmann, 1997, pp. 186–195.

[540] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: a multi-eesolution clustering approach for very large spatial databases," in *International Conference on Very Large Data*. Morgan Kaufmann, 1998, pp. 428–439.

[541] E. Smigiel, A. Belaïd, and H. Hamza, "Self-organizing maps and ancient documents," in *International Workshop on Document Analysis Systems*. Springer-Verlag, 2004, pp. 125–134.

[542] J. F. Rosenblatt, *Principles of neurodynamics*. Spartan Books, 1962.

[543] R. Xu, "Survey of clustering algorithms," *Neural Networks*, pp. 645–678, 2005.

[544] N. Barbuti and T. Caldarola, "An innovative character recognition for ancient book and archival materials: a segmentation and self-learning based approach," *Digital Libraries and Archives Communications in Computer and Information Science*, pp. 261–270, 2013.

[545] W. Boussellaa, A. Bougacha, A. Zahour, H. E. Abed, and A. Alimi, "Enhanced text extraction from Arabic degraded document images using EM algorithm," in *International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 743–747.

[546] B. Khelifi, N. Zaghden, R. Mullot, and M. A. Alimi, "Unsupervised categorization of heterogeneous text images based on fractals," in *International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[547] Y. Leydier, F. LeBourgeois, and H. Emptoz, "Serialized unsupervised classifier for adaptative color image segmentation: application to digitized ancient manuscripts," in *International Conference on Pattern Recognition*. IEEE, 2004, pp. 494–497.

[548] H. Zhang, J. Fritts, and S. Goldman, "Image segmentation evaluation: a survey of unsupervised methods," *Computer Vision and Image Understanding*, pp. 260–280, 2008.

[549] S. Wontaek, M. Agrawal, and D. Doermann, "Performance Evaluation Tools for zone Segmentation and classification (PETS)," in *International Conference on Pattern Recognition*. IEEE, 2010, pp. 503–506.

[550] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *International Journal of Computers and Communications*, pp. 27–34, 2011.

[551] E. Rendón, I. Abundez, C. Gutierrez, S. D. Zagal, A. Arizmendi, E. M. Quiroz, and H. E. Arzate, "A comparison of internal and external cluster validation indexes," in *Applications of Mathematics and Computer Engineering (AMERICAN-MATH/CEA 2011)*. World Scientific and Engineering Academy and Society (WSEAS), 2011, pp. 158–163.

[552] R. Sharan, A. Maron-Katz, and R. Shamir, "CLICK and EXPANDER: a system for clustering and visualizing gene expression data," *Bioinformatics*, pp. 1787–1799, 2003.

[553] J. M. Lewis, M. Ackerman, and V. R. D. Sa, "Human cluster evaluation and formal quality measures: a comparative study," in *Conference of the Cognitive Science Society*. Curran Associates, 2012, pp. 1870–1875.

[554] M. Halkidiand, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods: Part I," in *SIGMOD Record*. ACM SIGMOD Record, 2002, pp. 40–45.

[555] B. Mirkin, *Mathematical classification and clustering*. Dordrecht: Kluwer Academic Publishers, 1996.

[556] A. Rosenberg and J. Hirschberg, "V-measure: a conditional entropy-based external cluster evaluation measure," in *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*. Association for Computational Linguistics, 2007, pp. 410–420.

[557] B. C. M. Fung, K. Wangy, and M. Esterz, "Hierarchical document clustering using frequent itemsets," in *International Conference on Data Mining*. SIAM, 2003, pp. 59–70.

[558] A. H. Fielding and J. F. Bell, "A review of methods for the assessment of prediction errors in conservation presence/absence models," *Environmental Conservation*, pp. 38–49, 1997.

[559] Y. Zhao and G. Karypis, "Criterion functions for document clustering: experiments and analysis," Department of Computer Science, University of Minnesota, Tech. Rep. Technical report TR 0140, 2001.

[560] C. Y. Chiu, H. C. Lin, and S. N. Yang, "Texture retrieval with linguistic descriptions," in *Pacific Rim Conference on Multimedia: advances in Multimedia Information Processing*. Springer-Verlag, 2001, pp. 308–315.

[561] P. Howarth and S. Rüger, "Evaluation of texture features for content-based image retrieval," in *International Conference on Image and Video Retrieval*. Springer-Verlag, 2004, pp. 326–334.

[562] Y. L. Qi, "A relevance feedback retrieval method based on tamura texture," in *International Symposium on Knowledge Acquisition and Modeling*. IEEE, 2009, pp. 174–177.

[563] L. Paulhac, P. Makris, J. M. Grégoire, and J. Y. Ramel, "Human understandable features for segmentation of solid texture," in *International Symposium on Advances in Visual Computing: Part I*. Springer-Verlag, 2009, pp. 379–390.

[564] X. Zhang, P. Shen, J. Gao, X. X, D. Qi, L. Zhang, A. Xue, X. Liang, and X. Chen, "A license plate recognition system based on tamura texture in complex conditions," in *International Conference on Information and Automation*. IEEE, 2010, pp. 1947–1952.

[565] L. Nanni, A. Lumini, and S. Brahnam, "Survey on LBP based texture descriptors for image classification," *Expert Systems with Applications*, pp. 3634–3641, 2012.

[566] G. Zhao and M. Pietikäinen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *Pattern Analysis and Machine Intelligence*, pp. 915–928, 2007.

[567] L. Nanni and A. Lumini, "Local binary patterns for a hybrid fingerprint matcher," *Pattern Recognition Letters*, pp. 3461–3466, 2008.

[568] M. Prasad and A. Sowmya, "Multi-level classification of emphysema in HRCT lung images using delegated classifiers," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI*. Springer-Verlag, 2008, pp. 59–66.

[569] L. G. Brown and H. Shvaytser, "Surface orientation from projective foreshortening of isotropic texture autocorrelation," *Pattern Analysis and Machine Intelligence*, pp. 584–588, 1990.

[570] R. P. Heilbronner, "The autocorrelation function: an image processing tool for fabric analysis," *Tectonophysics*, pp. 351–370, 1992.

[571] H. C. Lin, L. L. Wang, and S. N. Yang, "Extracting periodicity of a regular texture based on autocorrelation functions," *Pattern Recognition Letters*, pp. 433–443, 1997.

[572] P. C. Chen and T. Pavlidis, "Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm," *Computer Graphics and Image Processing*, pp. 172–182, 1979.

[573] S. W. Zucker and D. Terzopoulos, "Finding structure in co-occurrence matrices for texture analysis," *Computer Graphics and Image Processing*, pp. 286–308, 1980.

[574] M. F. McNitt-Gray, N. Wyckoff, J. W. Sayre, J. G. Goldin, and D. R. Aberle, "The effects of co-occurrence matrix based texture parameters on the classification of solitary pulmonary nodules imaged on computed tomography," *Computerized Medical Imaging and Graphics*, pp. 339–348, 1999.

[575] A. Eleyan and H. Demirel, "Co-occurrence based statistical approach for face recognition," in *International Symposium on Computer and Information Sciences*. IEEE, 2009, pp. 611–615.

[576] A. A. Ursani, K. Kpalma, and J. Rosin, "Texture features based on Fourier transform and Gabor filters: an empirical comparison," in *International Conference on Machine Vision*. IEEE, 2007, pp. 67–72.

[577] D. Dunn, W. E. Higgins, and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions," *Pattern Analysis and Machine Intelligence*, pp. 130–149, 1994.

[578] D. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *Image Processing*, pp. 947–964, 1995.

[579] F. Bianconi and A. Fernández, "Evaluation of the effects of Gabor filter parameters on texture classification," *Pattern Recognition*, pp. 3325–3335, 2007.

[580] D. A. Clausi and M. E. Jernigan, "Designing Gabor filters for optimal texture separability," *Pattern Recognition*, pp. 1835–1849, 2000.

[581] S. Arivazhagan, L. Ganesanb, and S. P. Priyal, "Texture classification using Gabor wavelets based rotation invariant features," *Pattern Recognition Letters*, pp. 1976–1982, 2006.

[582] J. Rafiee, M. P. Schoen, N. Prause, A. Urfer, and M. A. Rafiee, "A comparison of forearm EMG and psychophysical EEG signals using statistical signal processing," in *International Conference on Computer, Control and Communication.* IEEE, 2009, pp. 1–5.

[583] M. Boukhris, A. A. Mohamed, D. D'Souza, M. Beck, N. E. B. Amara, and R. V. Yampolskiy, "Artificial human face recognition via Daubechies wavelet transform and SVM," in *International Conference on Computer Games.* IEEE, 2011, pp. 18–25.

[584] T. S. Lee, "Image representation using 2-D Gabor wavelets," *Pattern Analysis and Machine Intelligence*, pp. 959–971, 1996.

[585] Y. Liu and X. Zhou, "Automatic texture segmentation for texture-based image retrieval," in *International Conference on Multimedia Modelling.* IEEE, 2004, pp. 285–290.

[586] J. B. Abdeljelil, A. Kricha, and N. E. B. Amara, "Exploring a new wavelet in image processing," in *International Symposium on Industrial Electronics.* IEEE, 2008, pp. 780–785.

[587] A. Gavlasová, A. Procházka, and M. Mudrová, "Wavelet based image segmentation," in *Annual Conference Techincal Computing*, 2006.

[588] A. J. M. Traina, C. A. B. Castanon, and C. J. Traina, "MultiWaveMed: a system for medical image retrieval through wavelets transformations," in *Symposium on Computer-Based Medical Systems.* IEEE, 2003, pp. 150–155.

[589] S. W. Myint, N. S. N. Lam, and J. M. Tyler, "An evaluation of four different wavelet decomposition procedures for spatial feature discrimination in urban areas," *Transactions in GIS*, pp. 403–429, 2002.

[590] O. Svensson, K. Abrahamsson, J. Engelbrektsson, M. Nicholas, H. Wikström, and M. Josefson, "An evaluation of 2D-wavelet filters for estimation of differences in textures of pharmaceutical tablets," *Chemometrics and Intelligent Laboratory Systems*, pp. 3–8, 2006.

[591] G. V. de Wouwer, P. Scheunders, and D. V. Dyck, "Statistical texture characterization from discrete wavelet representations," *Image Processing*, pp. 592–598, 1999.

[592] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *Pattern Analysis and Machine Intelligence*, pp. 1186–1191, 1993.

[593] M. Unser, "Texture classification and segmentation using wavelet frames," *Image Processing*, pp. 1549–1560, 1995.

[594] K. Etemad and R. Chellappa, "Separability based tree structured local basis selection for texture classification," in *International Conference on Image Processing.* IEEE, 1994, pp. 441–445.

[595] N. Boughattas, H. Mahersia, and K. Hamrouni, "Rotation and scale invariant texture classification using wavelet transform and LBP operator," *International Review on Computers & Software*, 2013.

[596] E. Albuz, E. Kocalar, and A. A. Khokhar, "Vector-wavelet based scalable indexing and retrieval system for large color image archives," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1999, pp. 3021–3024.

[597] G. Sheikholeslami, A. Zhang, and L. Bian, "A multi-resolution content-based retrieval approach for geographic images," *GeoInformatica*, pp. 109–139, 1999.

[598] A. Busch, W. W. Boles, and S. Sridharan, "Logarithmic quantisation of wavelet coefficients for improved texture classification performance," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2004, pp. 569–572.

[599] J. Kautsky, J. Flusser, B. Zitová, and S. S̃imberová, "A new wavelet-based measure of image focus," *Pattern Recognition Letters*, pp. 1785–1794, 2002.

[600] M. K. Hu, "Visual pattern recognition by moment invariants," *Information Theory*, pp. 179–187, 1962.

Curriculum Vitæ

Name: Maroua MEHRI
Date of Birth: 23 Feb. 1986
Place of Birth: Sousse - Tunisia
Address: L3i - University of La Rochelle - Pascal Building, Office 123 - Michel Crépeau Str. - 17042 La Rochelle - France
Websites: `http://l3i.univ-larochelle.fr/Mehri-Maroua`
`https://sites.google.com/site/marouamehri/`
Email addresses: maroua.mehri@univ-lr.fr
maroua.mehri@gmail.com

Education and Work Experience

Oct. 2014 - Aug. 2015: Temporary lecturer and research assistant
Computer Science Department, Institute of Technology (IUT), University of La Rochelle - France

Oct. 2012 - Sep. 2014: Teaching assistant
Computer Science Department, IUT, University of La Rochelle - France

Nov. 2011 - Sep. 2014: Ph.D. in computer science
Laboratoire Informatique, Image et Interaction (L3i), University of La Rochelle - France
Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS), University of Rouen - France
About historical document image analysis
Under the supervision of Pr. Dr. Rémy MULLOT, Dr. Pierre HÉROUX and Dr. Petra GOMEZ-KRÄMER
Funded by the DIGIDOC ANR-10-CORD-020 research project
Committee members: Pr. Dr. Najoua ESSOUKRI BEN AMARA, Pr. Dr. Rolf INGOLD, Dr. Josep LLADÒS
Dr. Véronique EGLIN and Jean-Philippe MOREUX
Defended on 28 May 2015 with very honorable distinction with unanimous congratulations from the jury

Mar. 2011 - Aug. 2011: M.Sc. internship
VisAGeS - IRISA/INRIA - INSERM U746 - France
About image registration and data analysis in Deep Brain Stimulation (DBS)
Under the supervision of Dr. Pierre JANNIN, Dr. Florent LALYS and Dr. Mohamed Lassaad AMMARI
Committee members: Pr. Dr. Pascal HAIGRON, Pr. Dr. Xavier MORANDI,
Dr. Mohamed Ali MAHJOUB and Dr. Khaled KAÂNICHE
Defended on 13 September 2011 with an honors degree

Sep. 2010 - Feb. 2011: M.Sc. in signal/image processing
University of Rennes 1 - France
Electronics and Telecommunications (ET): Signal, Image, Embedded Systems and Automatic (SISEA)

Sep. 2010 - Feb. 2011: M.Sc. project
LTSI - INSERM U642 - France
About blind channel equalization in Single-Input Single-Output (SISO) or Single-Input Multiple-Output (SIMO) context
Under the supervision of Dr. Laurent ALBERA
Committee member: Dr. Amar KACHENOURA
Defended on 14 March 2011 with an honors degree

Sep. 2009 - Jun. 2010: M.Sc. in signals and communicating systems
National Engineering School of Sousse (ENISo), University of Sousse - Tunisia
Intelligent and Communicating Systems (SIC)

Jul. 2009 - Mar. 2010: Engineer developer

DOT Free and Open Source Software (DOTFOSS) company, Sousse - Tunisia
Development of IT security solutions

Feb. 2009 - Jun. 2009: Engineer internship

DotFOSS company, Sousse - Tunisia
Study and development of an integrated system for protecting borders of informatics networks
Under the supervision of Dr. Taha BEN SALAH and Dhiaeddine ROUIS
Committee members: Dr. Abdelaziz HAMDI and Dr. Mohamed Lassaad AMMARI
Defended on 16 June 2009 with an honors degree

Sep. 2006 - Jan. 2008: B.Sc. in applied computer engineering

ENISo, University of Sousse - Tunisia
Telecommunications and Industrial Networks (TRI)

Jul. 2008 - Aug. 2008: Engineer internship

DotFOSS company, Sousse - Tunisia
Management of IT security tools

Aug. 2007: Worker internship

Tunisie Telecom, Sousse - Tunisia
Telephone line connection

Jul. 2007: Enterprise education internship

Creative Media, Sousse - Tunisia
Web development

Sep. 2004 - Jun. 2006: Preparatory courses for national entrance examination to engineering education cycle

Preparatory School for Engineer Studies of Tunis (IPEIT), University of Tunis - Tunisia
Mathematics and Physics (MP)

Sep. 2000 - Jun. 2004: Secondary education cycle

Pioneer School of Sousse (LPS) - Tunisia
Mathematics

Journal Papers

1. **M. Mehri**, P. Héroux, P. Gomez-Krämer and R. Mullot, Texture Feature Benchmarking and Evaluation for Historical Document Image Analysis. Pattern Analysis and Machine Intelligence, IEEE, 2015 [submitted].

2. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, A Texture-based Pixel Labeling Approach for Historical Books. Pattern Analysis and Applications, Springer-Verlag, pages 1-40, 2015.

3. F. Lalys, C. Haegelen, **M. Mehri**, S. Drapier, M. Vérin and P. Jannin, Anatomo-clinical atlases correlate clinical data and electrode contact coordinates: Application to subthalamic deep brain stimulation, Journal of Neuroscience Methods, Elsevier Science, 212 (2), pages 297-307, 2013.

International Conference Papers

1. **M. Mehri**, P. Héroux, J. Lerouge, P. Gomez-Krämer and R. Mullot, A Structural Signature Based on Texture for Digitized Historical Book Page Categorization. *International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015 [accepted].

2. **M. Mehri**, P. Gomez-Krämer, P. Héroux, M. Coustaty, J. Lerouge and R. Mullot, A Bottom-up Method Using Texture Features and a Graph-based Representation for Lettrine Recognition and Classification. *International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015 [accepted].

3. J. C. Burie, J. Chazalon, M. Coustaty, S. Eskenazi, M. M. Luqman, **M. Mehri**, N. Nayef, J. M. Ogier, S. Prum and M. Rusiñol, SmartDoc: Smartphone Document Capture and OCR Competition. *International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015 [accepted].

4. **M. Mehri**, P. Héroux, N. Sliti, P. Gomez-Krämer, N. E. B. Amara and R. Mullot, Extraction of Homogeneous Regions in Historical Document Images. *In Proceedings of the* 10^{th} *International Conference on Computer Vision Theory and Applications (VISAPP)*, SciTePress, Berlin, Germany, 2015.

5. **M. Mehri**, N. Sliti, P. Héroux, P. Gomez-Krämer, N. E. B. Amara and R. Mullot, Use of SLIC superpixels for ancient document image enhancement and segmentation. *In Proceedings of the* 22^{nd} *Document Recognition and Retrieval (DRR), Part of the IS&T/SPIE 27th Annual Symposium on Electronic Imaging*, SPIE, San Francisco, CA, USA, 2015.

6. **M. Mehri**, M. Mhiri, P. Héroux, P. Gomez-Krämer, M. A. Mahjoub and R. Mullot, Performance Evaluation and Benchmarking of Six Texture-based Feature Sets for Segmenting Historical Documents. *In Proceedings of the* 22^{nd} *International Conference on Pattern Recognition (ICPR)*, IEEE, pages 2885-2890, Stockholm, Sweden, 2014.

7. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, A Pixel Labeling Framework for Comparing Texture Features: Application to Digitized Ancient Books. *In Proceedings of the* 3^{rd} *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, SciTePress, pages 553-560, Angers, France, 2014.

8. **M. Mehri**, P. Héroux, P. Gomez-Krämer, A. Boucher and R. Mullot, A Pixel Labeling Approach for Historical Digitized Books. *In Proceedings of the* 12^{th} *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, pages 817-821, Washington, DC, USA, 2013.

9. **M. Mehri**, P. Gomez-Krämer, P. Héroux and R. Mullot, Old document image segmentation using the autocorrelation function and multiresolution analysis. *In Proceedings of the* 20^{th} *Document Recognition and Retrieval (DRR), Part of the IS&T/SPIE 25th Annual Symposium on Electronic Imaging*, SPIE, San Francisco, CA, USA, 2013.

10. **M. Mehri**, F. Lalys, C. Maumet, C. Haegelen and P. Jannin, Analysis of electrodes' placement and deformation in deep brain stimulation from medical images. *In Proceedings of Medical Imaging: Image-Guided Procedures, Robotic Interventions and Modeling*, SPIE, San Diego, CA, USA, 2012.

International Workshop Papers

1. **M. Mehri**, N. Nayef, P. Héroux, P. Gomez-Krämer and R. Mullot, A Learning Texture-based Method for Enhancement and Segmentation of Historical Document Images. 3^{rd} *International Workshop on Historical Document Imaging and Processing (HIP)*, Tunis, Tunisia, 2015 [submitted].

2. **M. Mehri**, V. C. Kieu, M. Mhiri, P. Héroux, P. Gomez-Krämer, M. A. Mahjoub and R. Mullot, Robustness Assessment of Texture Features for the Segmentation of Ancient Documents. *In Proceedings of the* 11^{th} *International workshop on Document Analysis System (DAS)*, IEEE, pages 293-297, Tours, France, 2014.

3. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, Texture Feature Evaluation for Segmentation of Historical Document Images. *In Proceedings of the* 2^{nd} *International Workshop on Historical Document Imaging and Processing (HIP)*, ACM, pages 102-109, Washington, DC, USA, 2013.

National Conference Papers

1. **M. Mehri**, M. Mhiri, P. Gomez-Krämer, P. Héroux, M. A. Mahjoub and R. Mullot, Étude comparative de trois ensembles de descripteurs de texture pour la segmentation de documents anciens. *In Proceedings of the* 8^{th} *"Colloque International Francophone sur l'Écrit et le Document" (CIFED)*, pages 41-56, Nancy, France, 2014.

2. **M. Mehri**, V. C. Kieu, M. Mhiri, P. Héroux, P. Gomez-Krämer, M. A. Mahjoub and R. Mullot, Évaluation de la robustesse des descripteurs de texture pour la segmentation d'images de documents anciens. *In Proceedings of the* 8^{th} *"Colloque International Francophone sur l'Écrit et le Document" (CIFED)*, pages 25-40, Nancy, France, 2014.

3. V. C. Kieu, **M. Mehri**, V. Rabeux, N. Journet and M. Visani, Génération d'images semi-synthétiques de documents anciens à des fins d'évaluation de performances et d'apprentissage. *In Proceedings of the* 8^{th} *"Colloque International Francophone sur l'Écrit et le Document" (CIFED)*, pages 199-214, Nancy, France, 2014.

International and National Communications at Scientific Congresses Without Proceedings

1. **M. Mehri**, P. Héroux, P. Gomez-Krämer and R. Mullot, A structural method based on texture for ancient document image analysis. *ICDAR 2015 Doctoral Consortium*, Tunis, Tunisia, 2015 [accepted].
2. **M. Mehri**, P. Héroux, P. Gomez-Krämer and R. Mullot, Historical document image analysis: a structural approach based on texture. *Biennial Meeting of the French Research Group in Written Communication (GRCE)*, Paris, France, 2015.
3. **M. Mehri**, P. Gomez-Krämer, P. Héroux, A. Boucher and R. Mullot, Old document image segmentation using the autocorrelation function and multiresolution analysis. *Biennial Meeting of the French Research Group in Written Communication (GRCE)*, Paris, France, 2012.
4. F. Lalys, C. Haegelen, **M. Mehri** and P. Jannin, Anatomo-Clinical Atlases in SubThalamic Deep Brain Stimulation: correlating clinical data and electrode contacts coordinates. 3^{rd} *Annual Meeting of the ITMO Health Technologies*, Tours, France, 2011.

Teaching Activities

2014/2015: Temporary lecturer
Computer Science Department, IUT, University of La Rochelle - France

- Fundamental data structures and algorithms [1^{st} year] (C++): 16h - Tutorial classes & 32h - Practical tutorials
- Basics of object-oriented programming [1^{st} year] (Java): 16h - Tutorial classes & 50h - Practical tutorials
- Basics of object-oriented conception [1^{st} year]: 6h - Tutorial classes & 16h - Practical tutorials
- Description and project planning [1^{st} year]: 3.5h - Tutored project

2013/2014: Teaching assistant
Computer Science Department, IUT, University of La Rochelle - France

- Basics of object-oriented conception [1^{st} year]: 14h - Tutorial classes & 48h - Practical tutorials
- Databases [2^{nd} year]: 12h - Practical tutorials
- Analysis and numerical methods [1^{st} year]: 12h - Tutoring

2012/2013: Teaching assistant
Computer Science Department, IUT, University of La Rochelle - France

- Programming [1^{st} year]: 44h - Practical tutorials (C++ and Qt)
- Web programming [1^{st} year]: 12h - Practical tutorials (HTML5, CSS and JavaScript)

Other Activities

2015

- Participation in organizing ICDAR'15 Smartphone Document Capture and OCR Competition (SmartDoc-2015)
- Participation in organizing the 1^{st} digital innovation day in mutual/insurance field (INNOV15)

2014

- Supervision of a M.Sc. trainee working on the study of the development of a SLIC-based method for ancient document image enhancement and segmentation

– Presentation of the DIGIDOC project during the visit day of MAGELIS association and its corporate network at the University of La Rochelle (17 Avr. 2014)
– Participation in organizing the "Feature extraction" working group at L3i, Dounia Awad, Elodie Carel, Phuong Lai Hien, Maroua Mehri, Sophea Prum, University of La Rochelle, La Rochelle (03 Apr. 2014)
– Research stay at LITIS - University of Rouen (04 Nov. 2013 → 31 Jan. 2014)

2013

– Presentation of the "Comics Browser" tool during the science festival in La Rochelle (12 Oct. 2013)
– Supervision of a M.Sc. trainee working on the study of the robustness assessment of texture features for the segmentation of ancient documents

2012

– Research stay at LITIS - University of Rouen (16→26 Oct. 2012)
– Research stay at LITIS - University of Rouen (18→22 Jui. 2012)
– Research stay at LITIS - University of Rouen (23→28 Jan. 2012)

Technical and Personal Skills

Programming in C++ and Matlab, good basics of R, Java and Python
High writing and organizational skills
Very good communication, team work and interpersonal skills

Languages

Arabic	Native proficiency
French	Excellent proficiency
English	Good proficiency
German	Beginner

Hobbies and Interests

Volunteerism, hiking, traveling, new technologies and Web browsing

THESIS presented by **Maroua MEHRI**

Title:

Historical document image analysis: a structural approach based on texture

Abstract:

Over the last few years, there has been tremendous growth in digitizing collections of cultural heritage documents. Thus, many challenges and open issues have been raised, such as information retrieval in digital libraries or analyzing page content of historical books. Recently, an important need has emerged which consists in designing a computer-aided characterization and categorization tool, able to index or group historical digitized book pages according to several criteria, mainly the layout structure and/or typographic/graphical characteristics of the historical document image content.

Current systems for categorizing historical digitized book pages are based on several criteria, such as the textual content. However, these systems for performing the historical document image analysis tasks have poor performance due to many particularities of historical document images (e.g. large variability of page layout, noise and degradation, page skew, complicated layout, random alignment, specific fonts, presence of embellishments, variations in spacing between the characters, words, lines, paragraphs and margins, overlapping object boundaries, superimposition of information layers). Moreover, these systems are hindered by many issues related to the performance of the optical character recognition and retrospective conversion tools. In addition, they require burdensome and complex processing due to the mentioned particularities of historical document images.

Thus, the work conducted in this thesis presents an automatic approach for characterization and categorization of historical book pages. The proposed approach is applicable to a large variety of ancient books. In addition, it does not assume *a priori* knowledge regarding document image layout and content. It is based on the use of texture and graph algorithms to provide a rich and holistic description of the layout and content of the analyzed book pages to characterize and categorize historical book pages. The categorization is based on the characterization of the digitized page content by texture, shape, geometric and topological descriptors. This characterization is represented by a structural signature. More precisely, the signature-based characterization approach consists of two main stages. The first stage is extracting homogeneous regions. Then, the second one is proposing a graph-based page signature which is based on the extracted homogeneous regions, reflecting its layout and content.

Afterwards, by comparing the different obtained graph-based signatures using a graph-matching paradigm, the similarities of digitized historical book page layout and/or content can be deduced. Subsequently, book pages with similar layout and/or content can be categorized and grouped, and a table of contents/summary of the analyzed digitized historical book can be provided automatically.

As a consequence, numerous signature-based applications (e.g. information retrieval in digital libraries according to several criteria, page categorization) can be implemented for managing effectively a corpus or collections of books. To illustrate the effectiveness of the proposed page signature, a detailed experimental evaluation has been conducted in this work for assessing two possible categorization applications, unsupervised page classification and page stream segmentation. In addition, the different steps of the proposed approach have been evaluated on a large variety of historical document images.

Keywords: Digital libraries, Historical document image analysis, Segmentation, Categorization, Texture, Graph-based signature.

THÈSE présentée par **Maroua MEHRI**

Titre :

Analyse d'images de documents patrimoniaux : une approche structurelle à base de texture

Résumé :

Les récents progrès dans la numérisation des collections de documents patrimoniaux ont ravivé de nouveaux défis afin de garantir une conservation durable et de fournir un accès plus large aux documents anciens. En parallèle de la recherche d'information dans les bibliothèques numériques ou l'analyse du contenu des pages numérisées dans les ouvrages anciens, la caractérisation et la catégorisation des pages d'ouvrages anciens a connu récemment un regain d'intérêt. Les efforts se concentrent autant sur le développement d'outils rapides et automatiques de caractérisation et catégorisation des pages d'ouvrages anciens, capables de classer les pages d'un ouvrage numérisé en fonction de plusieurs critères, notamment la structure des mises en page et/ou les caractéristiques typographiques/graphiques du contenu de ces pages.

Les systèmes actuels de caractérisation et catégorisation des pages d'ouvrages numérisés s'appuient sur plusieurs critères relatifs au contenu textuel. Cependant, des performances insatisfaisantes ont été relevées en raison de divers problèmes, et qui sont liés aux particularités des documents anciens (e.g. une grande variabilité de la mise en page, des niveaux différents de dégradation et bruit, le défaut d'orientation, la complexité de la mise en page, des alignements non-conventionnels, les polices de caractères spécifiques, la présence d'ornements, les variations de l'espacement entre les caractères, mots, lignes, paragraphes et marges, la superposition de plusieurs couches d'information). En effet, leurs performances sont étroitement liées à celles des outils de reconnaissance optique de caractères et rétro-conversion. En outre, le traitement de ce type de documents peut s'avérer complexe et pénible en raison des particularités des documents anciens mentionnées ci-dessus, et ce, sans connaissances *a priori* sur la structure des mises en page ou les caractéristiques typographiques/graphiques du contenu de ces pages.

Ainsi, dans le cadre de cette thèse, nous proposons une approche permettant la caractérisation et la catégorisation automatiques des pages d'un ouvrage ancien. L'approche proposée se veut indépendante de la structure et du contenu de l'ouvrage analysé. Le principal avantage de ce travail réside dans le fait que l'approche s'affranchit des connaissances préalables, que ce soit concernant le contenu du document ou sa structure. Elle est basée sur une analyse des descripteurs de texture et une représentation structurelle en graphe afin de fournir une description riche permettant une catégorisation à partir du contenu graphique (capturé par la texture) et des mises en page (représentées par des graphes). En effet, cette catégorisation s'appuie sur la caractérisation du contenu de la page numérisée à l'aide d'une analyse des descripteurs de texture, de forme, géométriques et topologiques. Cette caractérisation est définie à l'aide d'une représentation structurelle. Dans le détail, l'approche de catégorisation se décompose en deux étapes principales successives. La première consiste à extraire des régions homogènes. La seconde vise à proposer une signature structurelle à base de texture, sous la forme d'un graphe, construite à partir des régions homogènes extraites et reflétant la structure de la page analysée. Cette signature assure la mise en œuvre de nombreuses applications pour gérer efficacement un corpus ou des collections de livres patrimoniaux (par exemple, la recherche d'information dans les bibliothèques numériques en fonction de plusieurs critères, ou la catégorisation des pages d'un même ouvrage). En comparant les différentes signatures structurelles par le biais de la distance d'édition entre graphes, les similitudes entre les pages d'un même ouvrage en termes de leurs mises en page et/ou contenus peuvent être déduites. Ainsi de suite, les pages ayant des mises en page et/ou contenus similaires peuvent être catégorisées, et un résumé/une table des matières de l'ouvrage analysé peut être alors généré automatiquement. Pour illustrer l'efficacité de la signature proposée, une étude expérimentale détaillée a été menée dans ce travail pour évaluer deux applications possibles de catégorisation de pages d'un même ouvrage, la classification non supervisée de pages et la segmentation de flux de pages d'un même ouvrage. En outre, les différentes étapes de l'approche proposée ont donné lieu à des évaluations par le biais d'expérimentations menées sur un large corpus de documents patrimoniaux.

Mots clés : Bibliothèques numériques, Analyse d'images de documents patrimoniaux, Segmentation, Catégorisation, Texture, Représentation structurelle à base de graphe.